

EXHIBIT A

EMC Data Manager

Software Reference

EDM Release 4.5.0

P/N 300-113-001-03

Rev C

EMC²

EMC Corporation, 171 South St., Hopkinton, MA 01748-9103

Corporate Headquarters: (508) 435-1000 (800) 424-EMC2

Fax: (508) 435-5374 Service: (800) SVC-4EMC

EMC Data Manager Software Reference

EDM Release 4.5.0

P/N 300-113-001-03 Rev C

March 2000

Copyright © 2000 EMC Corporation. All rights reserved.

EMC Corporation, 171 South St., Hopkinton, MA 01748-9103

This software/documentation contains proprietary information of EMC Corporation; it is provided under a license agreement that contains restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. EMC Corporation does not warrant that this document is error-free.

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data - General, including Alternate III (June 1987).

EMC², EMC, MOSAIC:2000, and Symmetrix are registered trademarks and EMC Enterprise Storage, EMC Enterprise Storage Network, EMC Enterprise Storage Specialist, EMC Storage Logic, The EMC Effect, The EMC Effect Alliance, The Enterprise Storage Company, Celerra, Connectrix, CopyPoint, EDM, E-Infostructure, FarPoint, InfoMover, PowerPath, SDMS, SRDF, SymmAPI, TimeFinder, Universal Data Tone, and Volume Logix are trademarks of EMC Corporation. Other trademarks are the property of their respective owners.

Contents

Preface	Who Should Use This Manual	xxx
	How to Get the Information You Need	xxx
	Online Help and Online Books	xxx
	Hardcopy Documentation	xxxi
	Symmetrix Path and Symmetrix Connect	xxxi
	Clients and Databases	xxxi
	Notes and Cautions	xxxii
	Technical Support	xxxii
	Reader's Comments	xxxii

PART I Basic Backup and Restore Concepts

Chapter 1 EDM Basics

EDM Overview	1-2
EDM Components	1-3
Network Backup and Restore	1-4
EDM Symmetrix Path	1-5

EDM Symmetrix Connect	1-6
Features	1-7
Mirroring	1-7
References	1-8
EDM Hardware	1-9
Disk Subsystem	1-9
Library Units	1-9
External Components	1-10
Remote Diagnostic Modem	1-10
Color System Console	1-10
EDM Software	1-11
Software Components	1-11
EDM Graphical User Interface	1-12
Starting the EDM GUI	1-13
Main Window	1-14
Backup Client Install Wizard	1-15
Backup Configuration Wizard	1-15
Backup Activity Wizard	1-15
Volume Management	1-16
Backup Configuration	1-17
Backup Report	1-18
Restore	1-19
System Monitor Configuration	1-20
HSM Client Installation	1-21
HSM Configuration	1-22

Chapter 2 Managing Your EDM System

Administering the System	2-2
Checking Media	2-3
Verifying Backup Completion	2-3

Reading the Daily Message File	2-4
Saving the EDM Backup Disaster Report	2-5
Check Compaction of Staging and Baseline Volumes	2-6
Running Procedures Automatically via Cron	2-6
Deleting Existing Entries in the crontab File	2-9
Backing Up Server Database Files	2-10

Chapter 3 Basic Backup and Restore Concepts

Client/Server Architecture	3-2
Key Processing Concepts	3-3
Scheduling	3-4
Automatic Scheduling	3-4
Custom Scheduling	3-4
Command Line Scheduling	3-5
Backup Activity Wizard	3-5
Concurrent Work Item Input	3-6
Balanced Scheduling	3-7
Rotation Period	3-7
Load Balancing	3-7
Multiplexed Storage	3-8
Nightly Backup Processing	3-10
Automatic Start	3-10
Client Processing	3-11
Backups of Changing Files	3-11
Storage of Backup Data	3-11
Cataloging of Backup Data	3-11
Restore Processing	3-12

Configuration Options	3-13
Key Configuration Options	3-13
Which Clients to Back Up?	3-14
What Data is Backed Up on Each Client?	3-15
When are Backups Scheduled?	3-15
Where is Backup Data Written?	3-16
Other Configuration Options	3-18
How are Backups Processed?	3-19
Permissions and User Modes	3-19
Catalogs and Backup Savesets	3-20
Reports and Logs	3-21
Reporting in the EDM GUI	3-22
Manual Operations	3-22

Chapter 4 Port Control

Overview	4-2
Understanding Port Control	4-3
Installing and Updating Client(s)	4-3
Restrictions	4-4
What is a Firewall?	4-5
EDM Firewall Assumptions	4-5
Firewall Requirements	4-8
Sample Firewall Configurations	4-9
Port Control Checklist	4-10
Setting Up the Firewall for Port Control	4-12
Enabling Port Control on the EDM	4-12
Portservices Files	4-14
localhost File	4-17

Installing Port Control on the EDM Client(s)	4-18
NOS Client Access Window	4-18
UNIX Client Install Method Window	4-18
Port Control Window	4-18
Making Changes to Port Control	4-20
To Change System Monitoring for Ping Errors	4-20
To Change the Default Port Range	4-20
An Alternative Method	4-20
To Enable Port Control with eb_server_config:	4-21
To Turn Off Portmapper on the EDM	4-22
An Alternative Method	4-22
To Turn Off Portmapper on Client(s)	4-22
To Set Portmapper Off for New Clients	4-22
To Disable Port Control for a Single Client	4-23
To Disable Port Control on the EDM and All of Its Clients	4-23

Chapter 5 How Backup and Restore Work

How Backup Works	5-2
Monitoring Active Backups	5-4
Start of Overall Backup	5-4
Client Access	5-4
Work Item Specification	5-5
Automatic Scheduling	5-5
Backup Activity Wizard	5-5
Client/Server Processing Methods	5-6
Standard Client Processing	5-6
High-Speed Client Processing	5-6
Server Processing	5-7
Filesystem Backup	5-7
Client Scans Filesystem	5-7

Database Backup	5-8
ACL Support	5-8
Backing Up Files with ACLs	5-8
Restoring Files with ACLs	5-9
Cross-Client Restore	5-9
Client ACL Commands	5-10
Client Pacing	5-10
Server Processes Attributes and Data	5-12
Catalog Processing	5-13
Server Database Update	5-14
Report and Log File Generation	5-14
Backup Completion Reports	5-14
Backup Failure Reports	5-15
Backup and Restore Logs	5-15
How Restore Works	5-16
Media Management	5-19
Expiring Backups	5-20
Deallocating Baseline Volumes	5-20

Chapter 6 Database Backup and Restore

Varieties of Database Backup	6-2
Database Backup Clients	6-3
EDM Symmetrix Path	6-3
EDM Symmetrix Connect	6-4
Various Database Backup Clients	6-6
Oracle Backup Client	6-6
EMC Backint Client for SAP R/3 Oracle Databases	6-6
Other UNIX Database Backup Clients	6-6
Microsoft Database Backup Clients	6-7

Database Network Backup Overview	6-7
Multiple Streams	6-8
Server-side Processing	6-8
Restores	6-9
User Interfaces	6-9
Configuring Backups to an Alternate Network	6-9
Database Pre-Discovery	6-10
EDM Symmetrix Path Overview	6-12
EDM Symmetrix Connect Overview	6-13
Applications	6-13
User Interfaces	6-13
Documentation	6-14
Raw Device Backups	6-14
Configurations	6-15
Mirrored Configurations	6-16
Non-Mirrored Configuration	6-17
EDM's Legacy "Offline" Database Backup Feature	6-17

Chapter 7 Basic Volume Management Concepts

Volume Management Overview	7-2
EDM Library Unit Manager	7-3
Volume Manager	7-3
Volume Catalog	7-4
Volume Life Cycle	7-4
Uncataloged	7-5
Unlabeled	7-6
Foreign	7-6
Expired	7-6
Erasing	7-7

Unverified Volume	7-7
How Volumes are Allocated	7-8
When a Volume is Allocated	7-8
Library Managers	7-12
Library Manager Configuration	7-12
Robotic Library Units	7-13
Offline and Offsite Library Managers	7-13
Ejecting a Volume	7-14

Chapter 8 How Volume Management Works

rmoper UNIX Group	8-2
Volume Management Processes	8-2
Volume Manager	8-3
Library Managers	8-4
Notify Daemon	8-4
Volume Management Startup	8-4
Manually Stopping and Restarting the vmdaemon	8-5
Using edmproc -restart	8-5
If an Error Occurs While Using edmproc -restart	8-7
Library Unit Operations	8-8
Inserting Media into Library Units	8-8
Importing a Volume	8-9
Inserting Cleaning Cartridges into Library Units	8-10
When Drives are Busy	8-11
Mounting and Dismounting Volumes	8-12
Mounting Volumes	8-12
Dismounting Volumes	8-13

Ejecting Media from a Library Unit	8-14
Ejecting Media Through the EDM GUI	8-15
Ejecting Media at the CLI	8-15
Drive Scheduling and Preemption	8-16
Drive Preemption	8-16
Verifying Priority in the Queue	8-16
LM_MAX_RESIDENT_TIME and	
LM_MIN_RESIDENT_TIME	8-17
Simultaneous Backup Example	8-17
Library Unit Inventories	8-18
Inventory Tables	8-19
How Inventories are Done	8-20
Delta Inventory	8-21
Barcode Inventories	8-21
Cleaning Tape Drives	8-22
Volume Allocation and Deallocation	8-23
How Volumes are Allocated	8-23
Volume Allocation Request	8-24
Mount Request	8-24
Volume Use	8-25
Duplicate Volume Sequence Numbers	8-26
When Volumes are Deallocated	8-27

Chapter 9 Media Duplication

The Media Duplication Process	9-2
Media Duplication Commands	9-3
Starting Duplication	9-3
Preparing a Backup Volume for Duplication	9-4
Selecting Append Mode or New Mode	9-5
Append Mode	9-5
New Mode	9-6

Configuring a Trail for Duplication	9-8
Initiating Manual Duplication at the CLI	9-8
Determining Duplicates of an Original Volume	9-10
Setting the Maximum Number of Concurrent Duplications	9-11
Configuring Concurrent Duplications in the EDM GUI	9-12
Configuring Concurrent Duplications at the CLI	9-12
Verifying the Status of a Duplication	9-13
If a Duplication Was Scheduled for an Offline Volume	9-14
Manually Disabling and Re-enabling Duplication	9-14
Disabling Duplication	9-15
Re-enabling Duplication	9-15
Pausing, Resuming, Canceling, or Removing a Duplication	9-16
Pausing Duplication	9-16
Resuming Duplication	9-17
Canceling Duplication	9-17
Removing a Failed Duplication from the Queue	9-18
If a Duplication Fails	9-19
Rescheduling a Failed Duplication	9-19
Rescheduling a Failed Duplication Through the GUI	9-20
Rescheduling a Failed Duplication at the CLI	9-21
Rescheduling Duplication of an Offline Original Volume	9-22
Rescheduling Duplication of a Single Volume for Archival Purposes	9-23

Restoring from Backup or Duplicate	9-24
If an Original Volume is Defective	9-25
Viewing Reports on Duplications	9-25
ebreport media Report	9-25
ebreport duplicate Report	9-27
Importing a Duplicate Volume	9-29
Importing a Duplicate Volume before the Original	9-29
Rejecting a Mount Request	9-29
Expiring a Duplicate Volume	9-30
Viewing Duplicate Expiration Dates	9-31

Chapter 10 Magnetic Disk Concepts

Expiration of Backups and Catalogs	10-2
Choosing Expiration Periods	10-3
Running Expiration	10-4
Filesystem Cleanup Script	10-4
Magnetic Disk Capacity	10-5
Rotation and Keep Catalog Periods	10-5
Calculating Actual Daily File Changes	10-7
Managing Disk Space	10-10
Distributing Catalogs	10-10
Reclaiming Magnetic Disk Space	10-13
Manually Expire Unneeded Catalogs	10-13
Other Options	10-14
Changing the Automatic Cleanup Script Defaults	10-15

PART II Hierarchical Storage Management (HSM)

Chapter 11 Basic HSM Concepts

- When Files Stage In and Out 11-2
- Filesystem Configuration and Maintenance 11-3
 - Staging Templates and Staging Trails 11-4
 - Deciding How Many Staging Templates to Create 11-5
 - Bitfiles and Client Stores 11-6
 - Deciding How Many Client Stores to Create 11-7
 - Preventing Redundant Backup of EDM Migration Client Data 11-9
 - Full Filesystems 11-10
 - Watermarks 11-10
 - Disk Utilization Zones 11-13
 - Sample Watermarks 11-15
 - Configuration Issues 11-18
 - Filesystem Limits 11-18
 - Stage-to-Tape Considerations 11-19
 - Self-Describing Media 11-22
 - Periodic Staging and Filesystem Delay 11-22
- File Control Properties 11-23
 - Listing and Changing File Control Properties 11-24
- Compaction of Staging Media 11-27
 - Administering Compaction 11-28
- Compacting Baseline Media 11-29
- Migration Reports 11-30
 - emfsreport and the Working Set 11-31
- Baseline Backup 11-33

Restaging Data 11-33

Backup Completeness 11-34

Chapter 12 How Migration Works

What Happens When You Enable Migration 12-2

Migration Configuration Database 12-2

How Stage-Out Works 12-4

The File Monitor Daemon (emfmd) 12-4

The Master Staging Daemon (emmasterd) 12-5

Candidate List Generation 12-5

What Happens When a File is Staged Out 12-6

How Stage-In Works 12-7

How the User-Level Commands Work 12-7

How the Network Migration Server Works 12-8

The EDM Migration Protocol 12-8

Network Migration Server Daemons 12-9

Network Migration Server Database 12-10

Client Stores 12-12

Bitfiles 12-13

How Compaction Works 12-14

Compaction Goals 12-14

Example 12-15

The Compaction Process 12-16

How Long Compaction Takes 12-18

Baseline Compaction 12-18

Deallocation and Reuse 12-18

Active Baseline Volumes 12-19

Recovering from Site Disasters 12-19

Chapter 13 HSM Command-line Tasks

HSM Commands	13-2
Test Staging	13-2
Set Up Periodic Staging	13-2
Tuning for Staging to Tape	13-3
Stage-to-Tape Tuning Parameters	13-3
Stage-to-Tape Tuning Procedure	13-4
Coordinating Automatic Procedures	13-4
Client (Periodic Stage Out)	13-5
Server (emvck)	13-6
Server (Periodic Stage Out)	13-6
Server (Compaction)	13-6
Server (Baseline Backup)	13-7
Server and Client (Backup)	13-7
Server (Backup Database)	13-7
Rotating Error Logs	13-8
Working with Individual Files	13-9
Staging Out Files	13-9
Staging In a Set of Files	13-9
Tagging a Set of Files for Future Stage Out	13-10
Locking a File on Magnetic Disk	13-11
Checking the Staging Configuration	13-12
Copying and Moving Data	13-13
Migrating Data from One Staging Trail to Another	13-13
Copying Files from One Filesystem to Another	13-14

Moving and Copying Files Between HSM Systems	13-14
Copying Files to Another HSM System (No Media)	13-15
Moving Files to Another EDM (Media Included)	13-15
Moving Files Between HSM Clients (Store Included)	13-16
Restoring a Staged Out File That Has Been Deleted	13-17
Copying Files to a Non-EDM System	13-17
Monitoring Storage Space and File Sizes	13-18
Maintaining Non-Stageable Filesystems	13-19
Managing Your Magnetic Disks	13-19
Run emfsreport	13-20
Choose the Desired Working Set in Days	13-21
Determine Additional Magnetic Disk Space Required	13-22
Reconfigure Magnetic Disk or Purchase More Disks	13-23
Populating Filesystems	13-23
Disabling Filesystem Staging	13-24
Temporarily Disabling Periodic Staging	13-25
Permanently Disabling Periodic and Demand Staging	13-25
Moving Staged Out Files to Another Filesystem	13-27
Compacting Staging Media	13-28
Administering Compaction	13-29
Compacting Baseline Media	13-30

Clearing Incomplete Bitfiles	13-31
Moving a Store to Another EDM Server	13-31
Gathering Migration Store Statistics	13-32
Checking a Network Client's Staging Configuration	13-33
Troubleshooting HSM	13-33
Restoring a Lost or Damaged Staging Volume	13-34
Restoring a Lost or Damaged Staging Volume (No Baseline Backup)	13-34
Restoring a Lost or Damaged Staging Volume (Base- line Backup is Enabled)	13-35
Restoring a Lost or Damaged Staging Trail	13-35
Restoring a Lost or Damaged Filesystem	13-36

PART III Logs and Reports

Chapter 14 Start of Backup and Related Processing

Backup Processing	14-2
Backup Activity Wizard	14-3
Automatic Nightly Processing	14-3
Automatic Scheduling	14-6
Custom Scheduling	14-6
Command Line Processing	14-7
Catalog Processing	14-9

Chapter 15 Message Logging

Message Logging Features	15-2
Syslog Message Files	15-2

Circular Log Files	15-4
Log File Rotation and Archival	15-4
Log Message Format	15-5
Default syslog Configuration File	15-6

Chapter 16 Backup Reports and Log Files

Report and Log Usage	16-2
Executing Reports from the EDM GUI	16-3
Report and Log Summaries	16-4
Backup Reports	16-5
Backup Media Reports	16-10
Backup Duplicate Reports	16-12
Duplicate Command Options	16-13
Sample Backup Duplicate Report	16-14
Backup History Reports	16-15
History Command Options	16-16
Sample Backup History Report	16-18
Backup Disaster Reports	16-19
Backup Baseline Reports	16-27
Backup Completion Reports	16-29
Backup Failure Reports	16-31
Backup Coverage Reports	16-32
Volume Reports	16-33

Log Files	16-35
Server Log Files	16-35
Local Client Log Files	16-37
Remote Client Log Files	16-37
Other Logs	16-37

PART IV Command Line Interfaces

Chapter 17 Configuring Library Managers

Using the lmconfig Utility	17-2
Listing Library Managers	17-3
Library Manager Name	17-3
SCSI Address	17-4
Installing Device Drivers	17-4
Updating Device Drivers	17-6
Removing Device Drivers	17-7
Configuring a Library Manager	17-8
Preparing for Configuration	17-8
Running lmconfig	17-9
If All Library Units Are Configured	17-10
If Media Is Found In Any Drive	17-10
The Configuration Process	17-11
Selecting the Cleaner Barcode Default	17-13
Viewing Log Files	17-13
Completing lmconfig	17-14
Deconfiguring a Library Manager	17-14

If You Have Trouble Configuring a Library Unit	17-17
If a Problem Occurs While Configuring Multiple Library Units	17-17

Chapter 18 Man Page Listing

Backup and Restore Man Pages	18-2
Volume Management Man Pages	18-6
HSM Man Pages	18-9

PART V Disaster Recovery

Chapter 19 Being Prepared for a System Disaster

Safeguarding Your Backup Media	19-2
Running and Saving Reports	19-3
MINIMAL Disaster Report	19-4
FULL Disaster Report	19-4
Redundant Backup Coverage	19-5
Configure Alternate Media Sets (Trailsets)	19-5
Media Duplication	19-5

Chapter 20 Recovering a Server from a Disk Failure

Steps to Restore a Server	20-4
Stop All Activity on the Server	20-5
Disable Activity	20-5
Reinstall Hardware and Software as Needed	20-6

Temporarily Reconfigure the Server	20-11
lmconfig	20-11
eb_server_config	20-13
Restore LOCAL_DATABASE Files	20-15
Reconfigure Library Units	20-23
Restore Catalogs and Backup Information Created After LOCAL_DATABASE Backup	20-23
Stop Backups	20-23
Run a Full Inventory and Import Uncataloged Volumes	20-24
Import Backup Catalogs and List Volumes	20-24
Restore Catalogs and Backup Information	20-26
Restore Data Created Before LOCAL_DATABASE Backup	20-28
Reenable crontab Entries	20-30
Restore Past Catalogs	20-30
Restore Missing Catalogs	20-32

Chapter 21 Recovering a UNIX Client from Disk Failure

Recovering a Client	21-2
Beginning Steps	21-2
For HSM Clients Only	21-3
For Backup and HSM Clients	21-4
For HSM Clients	21-5
For Both Backup and HSM Clients	21-6

PART VI Appendixes

Appendix A Directory Structure

Backup Server Directory Structure	<i>A-2</i>
/usr/epoch	<i>A-2</i>
/usr/epoch/EB	<i>A-4</i>
Bin Directory	<i>A-5</i>
Catalogs Directory	<i>A-5</i>
Client Directory	<i>A-6</i>
Config Directory	<i>A-8</i>
Db Directory	<i>A-9</i>
Locks Directory	<i>A-10</i>
Log Directory	<i>A-10</i>
Preconfig Directory	<i>A-10</i>
Server Man Directory	<i>A-10</i>
Table of Backup Server Directories and Files	<i>A-11</i>
Backup Client Directory Structure	<i>A-13</i>
Client Home Directory	<i>A-13</i>
Security Issues	<i>A-15</i>
Additional Client Software	<i>A-15</i>
Client EB_DB Directory	<i>A-16</i>
Client bin Directory	<i>A-16</i>
Client man Directory	<i>A-16</i>
Table of Backup Client Directories and Files	<i>A-17</i>
Volume Management Directory Structure	<i>A-18</i>
Library Manager Subdirectories	<i>A-18</i>
Volume Manager Files	<i>A-20</i>

HSM Directory Structure	A-21
HSM Configuration Database	A-21
Network HSM Server Database	A-22
Client Stores	A-23
Bitfiles	A-25

Appendix B EDM Backup Configuration File

General Coding Rules	B-2
Checking Your Changes	B-3
When Changes Take Effect	B-4
Summary of Fields	B-4
Server Fields	B-16
ebsvr	B-17
Client Backup Username	B-17
Backup Administrator Usernames	B-17
Authorized Backup List	B-18
Temporarily Disabling Backups for a Client	B-19
Authorized Recovery List	B-19
Authorized Cross-Recovery List	B-20
Disabling All Cross-Restores	B-22
Recovery Administrator List	B-23
Disabling Administrator-Level Restores	B-24
Maximum Simultaneous Clients	B-24
Use At Most n media-type Trails Concurrently	B-24
Limit Throughput To: nnn Per time	B-25
Specifying No Throughput Limit	B-27
Maximum Server backups.log File Size	B-27
Maximum Server recoveries.log File Size	B-28
Maximum Client backups.log File Size	B-28

Maximum Client recoveries.log File Size	B-29
Catalog Threshold to Force Level 0 Backup	B-29
Work Group Fields	B-31
Work Group Name	B-31
Filesystem Work Item Fields	B-32
When You Change a Work Item or a Filesystem	B-35
When You Stop Using a Work Item	B-35
Work Item Name	B-36
Client Name	B-37
Filespec to Back Up	B-37
Using the Block Form of the Syntax (Filespec Statement)	B-37
Backing Up the Local Client	B-38
Baseline Filespec	B-38
Migration Backup Tag	B-38
Exclusion Tag	B-40
Connection Via	B-40
Setting up the Network	B-41
Configuring the Work Items	B-41
Priority	B-41
Do Not Load Balance	B-43
Completeness	B-44
Level Map	B-45
Access Time Preservation	B-48
Maximum Files not Backed Up Before Forcing Full Backup	B-49
Database Work Item Fields	B-50
Database Work Item Name	B-51
Client Name	B-52
Filespec	B-52

Partition Spec	B-52
Database Type	B-52
Database Server	B-52
Database Name	B-52
Type	B-53
Exclusion Tag	B-53
Inclusion Tag	B-53
Access Time Preservation	B-53
Priority	B-54
Do Not Load Balance	B-54
Backup Client Initialization Command	B-54
Initialization Timeout	B-55
Backup Client Cleanup Command	B-55
Cleanup Timeout	B-55
Buffer Sizes	B-55
Backup Start Time	B-56
Level Map	B-56
PC Work Item Fields	B-56
Work Item Name	B-57
Connection Method	B-57
Filespec	B-57
Netware Username	B-58
Netware Encrypted Password	B-58
Netware Client TSA	B-58
Netware Client Target	B-58
Exclusion Tag	B-58
Backup Trailsets	B-58
Backup Trailset Name	B-60

Use Trail	B-60
Trail Name	B-61
Media Type	B-61
Backup Level	B-61
Maximum Number of Concurrent Clients	B-62
Use Level B1 / B2 For Baseline Backups	B-64
Keep Backups	B-65
Keep Duplicates	B-67
Keep Backup Catalogs	B-68
Keep Saveset Records	B-68
Backup Catalog Delta Level	B-69
Backup Template Fields	B-70
Backup Template Name	B-71
Work Group List	B-73
Begin Trailset Rotations	B-73
Rotation Period	B-74
Primary and Alternate Trailsets	B-75
Logging Level	B-77
Server Log File	B-78
Backup Completion Script	B-79
Backup Failure Script	B-80
Do All Baseline Backups Before Normal Backups	B-80
Recreate Baseline if Needed	B-81
Schedule	B-82
Standard vs. Full-During-Weekends Rotations	B-83
Specifying Backup Shifts	B-84
Scheduling Custom Backups (Level n on Days...)	B-84
Startup Parameters	B-86

Appendix C Volume Management Configuration Files

- Volume Manager Configuration File *C-2*
- Library Manager Configuration Files *C-9*
 - Library Manager Naming Convention *C-9*
 - Library Manager Subdirectories *C-10*
 - Library Manager Configuration Parameters *C-11*
 - The LM_INLET_IGNORE_ON_OPEN
Parameter *C-19*
 - Offline and Offsite Library Managers *C-20*
 - Offline Library Manager *C-20*
 - Offsite Library Manager *C-21*

Appendix D findxcpio Directives

- Work Item Directive *D-2*
- Logical Operators *D-2*
- Macros *D-3*
 - Compound Macros *D-5*
- Syntax to do Back Up *D-7*
- Syntax to not Back Up *D-8*
- Evaluation Shortcuts *D-11*
 - False and *D-11*
 - True or *D-12*

Glossary

Index

Preface

The *EMC Data Manager Software Reference* manual is the core document that is provided to customers who purchase either an EMC Data Manager (EDM) or EDM with the Hierarchical Storage Management (HSM) Option.

This manual provides comprehensive reference information about Backup, Restore, Volume Management, and optional HSM software. This manual introduces basic EDM software concepts and describes the programs, reports, and log files that you use in conjunction with the EDM interface. It also provides overall disaster recovery instructions for the EDM server.

This manual describes the following:

- Basic Backup and Restore Concepts
- Volume Management and Duplication Concepts
- Basic HSM Concepts
- Logs and Reports
- Command Line Interfaces
- Disaster Recovery
- Configuration Files

Who Should Use This Manual

This manual is intended for system administrators who are responsible for administering and operating an EMC Data Manager (EDM) or EDM with HSM Option. You should be familiar with UNIX system administration, understand your network environment, and understand the requirements of the various groups that you serve.

How to Get the Information You Need

Information about the EMC Data Manager is available through the graphical user interface (GUI) online help and online books, as well as hardcopy documentation, as described in the following sections.

Online Help and Online Books

The EDM graphical user interface Help facility provides detailed information and procedures for managing the EDM server, its clients, and library units.

To access online help, log in to the EDM. Enter **edm** to display the EDM Main window, then click on the Help button. The Help facility includes context-sensitive help for input fields, window areas, options, and buttons. Online Help also includes step-by-step instructions for tasks that you perform to manage and monitor the EDM server.

Also available are online versions of selected manuals. To access these online books, select Help in the EDM Main window menu bar, then Online Books.

Hardcopy Documentation

The following documentation is available for use with EMC Data Manager (EDM) or EDM with HSM Option.

- *EDM Software Reference* (P/N 300-113-001)
- *EDM Software Release Notes* (P/N 300-113-004)
- *EDM Storage Devices* (P/N 300-113-002)
- *EDM Server Error Messages* (P/N 300-113-014)

Symmetrix Path and Symmetrix Connect

The *EDM Symmetrix Path User Guide* (P/N 300-113-007) provides information for configuring, backing up, and restoring data using EDM Symmetrix Path.

The *EDM Symmetrix Connect User Guide* (P/N 300-113-005), the *EDM Symmetrix Connect Quick Reference Card* (P/N 300-113-006), and the *Symmetrix Connect Checklist* (P/N 300-113-011) provide information for configuring, backing up, and restoring data using EDM Symmetrix Connect.

Clients and Databases

The following EMC Data Manager supplemental guides provide installation and configuration instructions for setting up the EDM backup clients. Release notes are also available.

- *NetWare Backup Client* (P/N 300-114-001)
- *OS/2 Backup Client* (P/N 300-118-001)
- *Windows NT Backup Client* (P/N 300-119-001)
- *OpenVMS Backup Client* (P/N 300-122-001)
- *Oracle Backup Client* (P/N 300-115-001)
- *Sybase Backup Client* (P/N 300-116-001)
- *Informix Backup Client* (P/N 300-117-001)
- *EMC Backint for SAP R/3 System* (P/N 300-120-001)
- *Windows NT SQL Server Backup* (P/N 300-121-001)
- *Windows NT Oracle Backup Client* (P/N 300-115-003)

- *Windows NT Exchange Backup Client* (P/N 300-119-003)
 - *Windows NT Lotus Notes Backup Client* (P/N 300-119-005)
-

Notes and Cautions

Notes provide clarification or additional important information.

Note: It calls your attention to an operating procedure, practice, condition, or similar situation which is important to highlight.

Cautions are used to indicate the presence of a hazard.

CAUTION: It calls your attention to an operating procedure, practice, condition, or similar situation. Failure to observe a caution can result in minor personal injury, damage to a program, device, system, or loss of data.

Technical Support

If you need technical assistance with the EDM system, contact the EMC support hot line at:

1 800 SVC-4EMC (1 800 782-4362)

If you are located outside the United States or Canada, contact the nearest EMC office for assistance.

Reader's Comments

We welcome your comments on this documentation.

Send e-mail to:

doc_comments@mil.emc.com

Please include the part number and title of the manual.

Part I

Basic Backup and Restore Concepts

1 EDM Basics

EMC Data Manager (EDM) is an integrated solution for unattended backup of data over your network, EDM Symmetrix Path, or EDM Symmetrix Connect.

The EMC Data Manager is a full-featured, client/server backup system that offers fast and reliable backup processing with minimal operator intervention.

With its graphical user interface (GUI) and built-in intelligent scheduling, EDM Backup software fully manages the complete backup process to ensure maximum protection of all data in a client/server environment.

This chapter describes the following topics:

- EDM Overview
- EDM Hardware
- EDM Software
- EDM Graphical User Interface

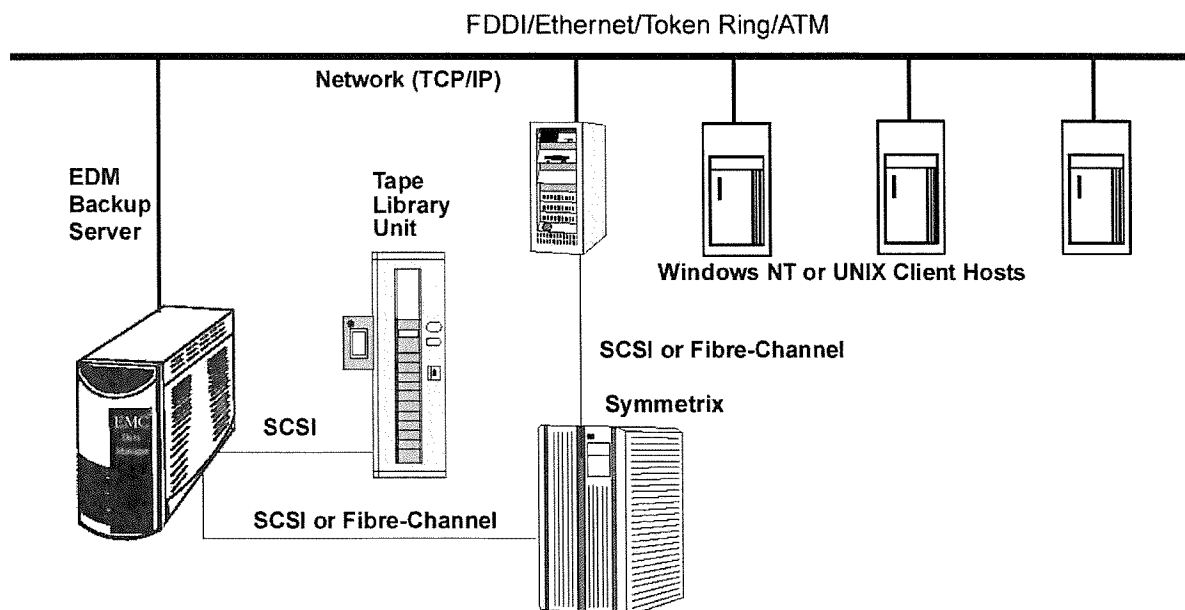
EDM Overview

EMC Data Manager backs up data from magnetic disk to the magnetic tape, backing up both databases, tablespaces, and filesystems. EDM writes the backup data to tape cartridges within one or more tape library units attached to the EDM. In so doing, EDM gives you the ability to perform restores of data lost by physical disk failure and restores of data lost by removal (logically) from the disk.

EDM backs up data from both Symmetrix-attached computers and from networked computers. For Symmetrix-attached computers, EDM's high-performance Symmetrix Connect and Symmetrix Path options offer rapid transfer of large amounts of backup data over Fibre-Channel or SCSI cabling between the Symmetrix and EDM. For computers that are not attached to a Symmetrix, EDM transfers backup data over the network.

Figure 1-1

EDM Symmetrix and Network Backup Environment



With its Hierarchical Storage Management (HSM) option, EDM migrates less-used filesystem data from magnetic disks to optical disks located in attached optical library units. One benefit of HSM is to reduce full backup loads.

The EDM Backup software maintains catalogs of the backups on disk storage, and restores filesystem and database data from magnetic tape to the client.

The EDM software also controls the operation of robotics, drives, and cartridges located inside tape library units. State-of-the-art product features include a multi-threaded library manager, media duplication capability, and ATL StorLink support.

A graphical user interface enables you to configure and manage tape operations, and network backups and restores of backed up files and databases.

EDM Components

The EDM system includes both the hardware and software components that are needed to backup and restore data. The EDM hardware consists of:

- an EDM cabinet containing: a server unit, magnetic disks for online catalogs, and optional internal tape library unit
- a color system console, keyboard, mouse, and modem
- attached external tape library units (DLT, 8mm) and/or optical library units (EO, WORM)

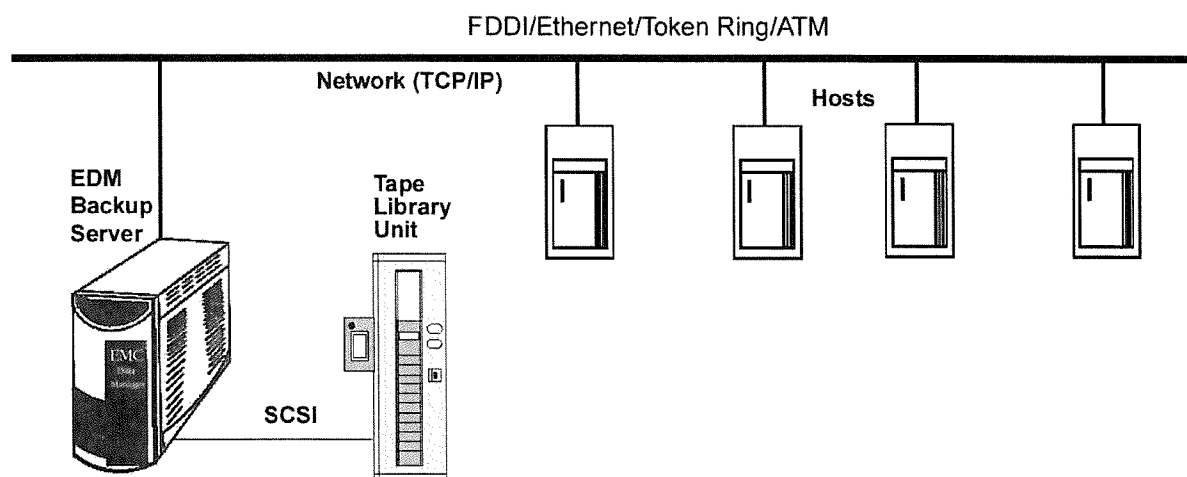
EDM software components include a Sun Solaris operating system and EDM Backup software (which includes system monitoring support software). The software is installed by EMC and vendor factory and/or EMC Customer Service personnel. Optionally included are Sun ATM, FDDI, Ethernet, and/or Token Ring card device driver software, and EMC Hierarchical Storage Management (HSM) software.

Network Backup and Restore

Over the network, EDM can back up from a broad selection of UNIX and Windows NT platforms and PC clients acting as file servers, database management systems, and application workstations. Computer platforms in the EDM network environment include the EDM itself and the other hosts that you configure as network backup clients. (Refer to the *EMC Data Manager Software Release Notes* for a current list of supported clients for network and direct connect backup.) Data from each host is streamed over the TCP/IP network to the EDM, which writes the data to tape library units (TLUs) that are attached to it through SCSI connections. Figure 1-2 illustrates an EDM network environment.

Figure 1-2

EDM Network Backup and Restore Environment



The EDM software maintains catalogs of the backups to enable easy interactive restoring whenever necessary. The backup catalogs are stored online on the disk subsystem in or external to the EDM cabinet.

EDM performs online, full and incremental backups of filesystems. Full backups copy the entire filesystem, while incrementals just copy those files that have been added or changed since the previous backup. (Incremental backups also note down any files that have been deleted.) EDM can back up databases online or offline, as applicable to particular database systems.

As EDM system administrator, you can automate filesystem backups and you can also initiate backups and restores on demand from the EDM. Also, you can optionally configure the restore feature to enable some or all of your workstation and fileserver users to restore their own files and filesystems without your assistance.

In most cases, database backups can be either client-initiated or EDM-initiated. With client-initiated backups, the database administrators manage their own database backups and restores from the database server (the client system to the EDM). With EDM-initiated backups, you can configure automated schedules as well as initiate backups on demand. In a few cases, database backups can only be EDM-initiated, and in a few cases, only client-initiated. Similarly, in many cases, database restores can either be EDM-initiated or client-initiated. But in some cases, database restores can be EDM-initiated only, and in other cases, client-initiated only.

EDM Symmetrix Path

With EDM Symmetrix Path, you can back up and restore many of the same types of databases and filesystems that you can over the network, but the data is transferred through a Symmetrix instead of over the network. (Refer to the *EMC Data Manager Software Release Notes* for a current list of supported client platforms.) The EDM Symmetrix Path option backs up large databases and filesystems through the Symmetrix cable connections, thereby bypassing the network's limited speed and bandwidth.

With this methodology, the EDM server and a client system are both cabled to a Symmetrix and the Symmetrix itself acts as the network. A few small devices on the Symmetrix are designated as transport paths for configuration purposes, while the actual data transport is generally handled by cache memory. Various device configurations and host mapping combinations are possible to provide different performance characteristics and degrees of flexibility.

Symmetrix Path can backup data residing on the locally attached disks of a Symmetrix-attached computer, as well as data on Symmetrix disks. Refer to the *EMC Data Manager Symmetrix Path User Guide* for more information about this feature.

EDM Symmetrix Connect

The EMC Data Manager Symmetrix Connect feature permits a high speed online or offline backup and restore of a client's very large UNIX filesystem, Windows NT, Oracle7, Oracle8, or Oracle8i database (or a copy) that resides on one or more Symmetrix systems.

EDM Symmetrix Connect backs up data residing on Symmetrix disks, either on so-called mirrored volumes — TimeFinder Business Continuance Volumes (BCVs) or Symmetrix Remote Data Facility (SRDF) target (R2) volumes — or on unmirrored volumes — TimeFinder standard devices (STDs) or SRDF source (R1) volumes.

Symmetrix Connect provides EDM-specific interfaces to back up database and filesystem data on UNIX and Windows NT clients. For UNIX systems (IBM, Compaq, HP, Sequent, and Sun), the EDM Oracle Application Interface backs up Oracle7/8/8i databases and filesystems. On Windows NT clients, various EDM-specific interfaces enable backup of filesystems, Oracle Backup, SQL Server, and Exchange.

Symmetrix Connect also backs up UNIX Oracle databases through two standard interfaces. One is Oracle8's Recovery Manager (RMAN) using its Proxy Copy feature. The other is the SAP R/3 System SAPDBA utility through EMC's SAP-certified interface client: EMC Backint.

Features

Symmetrix Connect offers these features:

- high speed backup throughput to tape media
- Oracle database and tablespace-level online and offline backup capability, plus data file-level backup if using RMAN Proxy Copy
- data file-level restore (plus disaster recovery) capability
- minimal impact on your local area network
- minimal impact on the client (whose database is being backed up)
- UNIX filesystem backup for all supported clients
- Oracle filesystem backup for all supported clients
- Windows NT support for Oracle Backup, Exchange Server Backup, SQL Server Backup, and filesystems
- SAP R/3 Symmetrix Connect support for Oracle databases on UNIX clients database running EMC Backint
- multiple simultaneous database support for UNIX clients
- PowerPath interoperability
- GUI client configuration and backup capability
- enhanced GUI System Monitoring Support features

Mirroring

If you are using a mirroring facility, Symmetrix Connect backups take advantage of three Symmetrix features to provide high performance database backup with minimal impact on your database server (host):

- Symmetrix Remote Data Facility (SRDF)
- Symmetrix TimeFinder (or SMMF™)
- Symmetrix Remote BCV in an SRDF Configuration

In addition, the multipath disk access feature is Symmetrix's ability to allow multiple (local and remote) hosts to read and write data to and from the same disk at the same time.

Note: Hierarchical Storage Management (HSM) is not supported with EMC Data Manager Symmetrix Connect software.

References

For more information on Symmetrix Connect, refer to:

- The *EMC Data Manager Symmetrix Connect User Guide* for information on using this feature with the EDM Oracle Application Interface for UNIX clients and Filesystem Application for UNIX clients.
- The following guides for information on Symmetrix Connect with the EDM-specific interfaces to NT filesystems and databases:
 - *EMC Data Manager Windows NT Backup Client*
 - *EMC Data Manager Windows NT Oracle Backup Client*
 - *EMC Data Manager Windows NT SQL Server Backup Client*
 - *EMC Data Manager Windows NT Exchange Backup Client*.
- The *EMC Data Manager Oracle Backup Client* guide for information on using Symmetrix Connect with RMAN Proxy Copy (on UNIX clients).
- The *EMC Data Manager EMC Backint* guide for information on using Symmetrix Connect with the SAP R/3 System's SAP Tools (on both UNIX and Windows NT clients).

EDM Hardware

You can purchase an EDM in several configurations (models) and with various system upgrades.

The server unit can be configured with multiple SBus cards; their use depends on your SCSI peripheral and networking requirements. SBus board options include Ultra-SCSI (differential), EDM-Fibre (Fibre-Channel), ATM, Quad Ethernet, Fast Ethernet, Token Ring, and FDDI.

Disk Subsystem

In addition to a Power Distribution Unit (PDU) and a server unit, most EDM cabinets contain a disk subsystem that stores catalogs and contains operating system and application software.

If you have a SPARCserver 1001E or Ultra Enterprise 4000 (or 4500) unit, a Sun disk subsystem is provided for the storage of backup catalogs, EDM Backup software, Solstice, and other files. An Ultra Enterprise 3000 (or 3500) system contains six or more internal disk drives that function in a similar manner.

The disk subsystem catalogs all files that were backed up by EDM. The disk subsystem is required in network-only (non-Symmetrix) backup environments.

For concepts on the use capacities of the various catalog disk subsystems, refer to "Magnetic Disk Capacity" on page 10-5.

Library Units

EDMs are equipped with tape library units (DLT, DTF, HITC, 8mm) and/or optical library units (EO or WORM) that provide secondary online storage and perform unattended backup of user data. Each library unit holds a robotics or picker system, one or more drives (tape or optical), and media that enables you to store from tens of gigabytes to tens of terabytes of data.

Refer to *EMC Data Manager Storage Devices* for more information about supported library units.

External Components

In addition to the EDM cabinet(s) and any external library units, your EDM comes with the following hardware:

- Remote Diagnostic Modem
- Color System Console

Installation of Symmetrix systems is handled separately.

Remote Diagnostic Modem

A remote diagnostic modem (RDM), external to the cabinet and equipped with RS-232 cable, enables dial-in. The modem must be connected to a dedicated telephone line.

The modem and the Remote System Monitoring (RSM) function enable the EDM to notify the EMC Customer Service Database about general system information and any problems with the EDM system. The modem and RSM also enable remote-user dial-in by EMC personnel to query the EDM.

The EMC Customer Service Database is the service management software that Customer Engineers, Customer Service Technicians, and Product Support Engineers use to log service activity, track field service inventory, download files, and to look up other important customer information.

Color System Console

The Sun system console has a color, high-resolution, bit-mapped display and includes a keyboard, and a three-button optical mouse.

The color system console uses the Common Desktop Environment (CDE) for displaying the EDM GUI. It enables GUI-based system administration of the EDM as well as access to applications that run on network accessible hosts.

EDM Software

The EDM Backup software with client/server architecture supports Symmetrix Connect and Symmetrix Path as well as standard network backups. The server software is located on the EDM unit. Client software is located on the EDM and on the other hosts within the network architecture.

For more information, refer to “Client/Server Architecture” on page 3-2.

Software Components

EDM software consists of the following components:

- Backup — provides centralized backup and restore services for the server and clients on the network. The EDM’s backup server software runs on the EDM server. The client software runs on the server as a local client and on each networked (remote) client. To understand network backup and restore, review Chapter 3 “Basic Backup and Restore Concepts,” Chapter 5 “How Backup and Restore Work,” and Chapter 6 “Database Backup and Restore.”
- Volume Management — provides integrated media and library management for the EDM server. The volume management software keeps track of all media that is known to the server, whether it is in a library unit or an offline or offsite location. For an understanding of volume management, review Chapter 7 “Basic Volume Management Concepts,” Chapter 8 “How Volume Management Works,” and Chapter 9 “Media Duplication.”
- HSM (Hierarchical Storage Management) — is an option that is available with EDM network backups. This option provides HSM for the EDM system, both for the local server and for networked clients. For an understanding of HSM, review Part II, “Hierarchical Storage Management.”

- **System Monitoring Support** — enables Customer Service and/or system administrators to configure and receive notification of serious system problems that prevent successful completion of the backups or related issues and system status. When configured with RSM software, notifications can be sent to EMC's Customer Service Database system and to designated email recipients. SNMP traps can be generated and Tivoli event messages issued. Use the System Monitor Configuration GUI to configure it, and online Help to learn more about its features.
- **Client** — is available for a wide range of PC, Windows NT, and database clients.

EDM Graphical User Interface

For network applications, use the EMC Data Manager graphical user interface (GUI) to install and configure both backup and HSM clients, manage media volumes, monitor library units on the EDM, configure backups, restore data, monitor backup activity, and display reports. Also, use it to set up the migration of data if you have the HSM option.

Note: The software for HSM must first be installed directly on the client.

Context sensitive online Help is available throughout the EDM user interface. Online versions of hardcopy books are available through the EDM button in Online Help.

Note: The name of the EDM (server) is displayed in the title bar of each EDM window.

For Symmetrix Connect and Symmetrix Path, use the GUI to install clients, configure clients, manage media volumes, and monitor library units on the EDM. You can use the GUI to perform network restores and Symmetrix Path restore operations. You cannot use the GUI for Symmetrix Connect restores.

Starting the EDM GUI

To start the EDM GUI on the EDM server, log in to the EDM. (If you want to configure and manage backups, log in as root or enter `su -` from your user account. The “-” is required.)

Then set your environment and enter the **edm** command:

```
# setenv DISPLAY nodename:0.0
# edm &
```

Starting the EDM GUI Remotely

From any EDM or client, you can remotely launch the EDM GUI from any other EDM and display it on any EDM or client.

1. Set your environment to designate the machine on which you want to display the EDM GUI.

```
client# setenv DISPLAY nodename:0.0
```

2. Enter **edmremote** with the remote EDM name, and enter the root password when prompted.

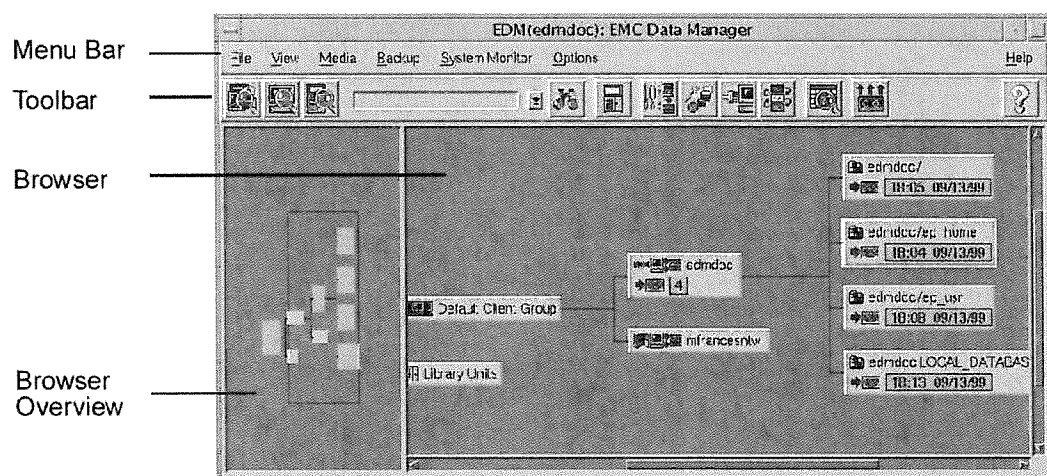
```
client# edmremote edm_name
Please enter password for <root> on <edm_name>
Password:
edmremote - Remote GUI launch complete on
<edm_name>. DISPLAY = nodename:0.0
```

Refer to the **edmremote** man page for variations.

Main Window

The EDM Main window provides a central location for viewing and managing clients and library units connected to the EDM server, and for monitoring and reporting on backup activity.

The main viewing area contains the browser and browser overview, as shown.



You can access the following EDM functions from the menu bar, tool bar, or a pop-up menu in the browser.

- Backup Client Install Wizard
- Backup Configuration Wizard
- Backup Activity Wizard
- Volume Management
- Backup Configuration
- Backup Report
- Restore
- System Monitor Configuration
- HSM Client Installation¹
- HSM Configuration

Note: The HSM Client Installation and Configuration windows do not appear unless you have the HSM option installed on your EDM.

Backup Client Install Wizard



Click on this icon to install a backup client. The Backup Client Install Wizard leads you step by step through the install process:

Refer to EDM online help for more information about this wizard.

When you complete the installation, a window opens and asks if you want to configure a simple backup.

Backup Configuration Wizard



The Backup Configuration Wizard enables you to configure a network, Symmetrix Path, or Symmetrix Connect backup of filesystems or a database. This wizard supports database backups for Oracle, Informix, Exchange, SQL Server, Lotus Notes, and Sybase. The Backup Configuration Wizard leads you step by step through the configuration process. Refer to EDM online help for more information about this wizard.

When you complete the configuration, backups can start, either automatically if you choose that option in the wizard, or manually using either the Backup Activity Wizard or the command line interface.

Backup Activity Wizard



The Backup Activity Wizard enables you to start new, queued, or failed backups, stop running backups, or manage the backup queue.

In the Wizard panels you select a backup operation, select the objects that you want to operate on, choose backup options, and confirm your actions. Then from the Main window, you can monitor the progress of the backup operation that you initiated. (Refer to EDM online help for more information about this wizard.)

Note: You must have root privileges or be an EDM Backup Administrator to use the Backup Activity Wizard.

Volume Management



Click on this icon to display the Library Unit Manager window. In this window, you manage and monitor media, library units, and drives. You can label media (volumes), take an inventory of tapes or optical disks in a unit, find specific tapes or disks, create and save customized settings in the media list, restart failed media duplications, and perform other related tasks. (Refer to EDM Online Help for more information about these options.)

Volume management also alerts you when operator intervention is needed. For example, if additional media is needed to complete a backup, a window appears that indicates the volume(s) that are needed to complete the operation. For more information about volume management, refer to Chapter 7 “Basic Volume Management Concepts,” Chapter 8 “How Volume Management Works,” and Chapter 9 “Media Duplication.”

Library Units and Drives

Media List

Buttons

Tabs

EDM(stealth): Library Unit Manager

Library Unit	Slot	Sequence	Barcode	Allocation Time	Current Use	Data Written	Identical to
stl_3264.0	1	Cleaner	2LW01				
stl_3264.0	1		3F7676				
stl_3264.0	3		36V005				
stl_3264.0	2	+E3	36V554	12/13/93 12:04:03	Backup	69.49 MB	
stl_3264.0	5	+E6	3F7924	12/13/93 18:03:36	Backup	3.71 MB	
stl_3264.0	9	+E6	36V632	12/16/93 18:18:57	Backup	1.74 MB	
stl_3264.0	8	+E0	36V003	12/16/93 18:18:58	Backup	601.01 MB	
stl_3264.0	0	+E3	36V007	12/07/93 13:30:18	Function...	27.59 MB	3F388*
stl_3264.0	7	+E1	3FV895	12/02/93 16:29:23	Backup	25.29 MB	46X007
stl_3264.0	6	+E1	36V549	12/14/93 13:41:27	Backup	66.23 MB	

10 displayed, 1 selected, 0 scheduled.

Inventory | Search | Information | Utilities | Label | Eject | Filters | Columns

Available Columns: Comment, Dr, Name, Opens, Schedules For

Displayed Columns: Library Unit, Slot, Sequence #, Barcode, Allocation Time

Sort: Primary: Library Unit, Secondary: Slot

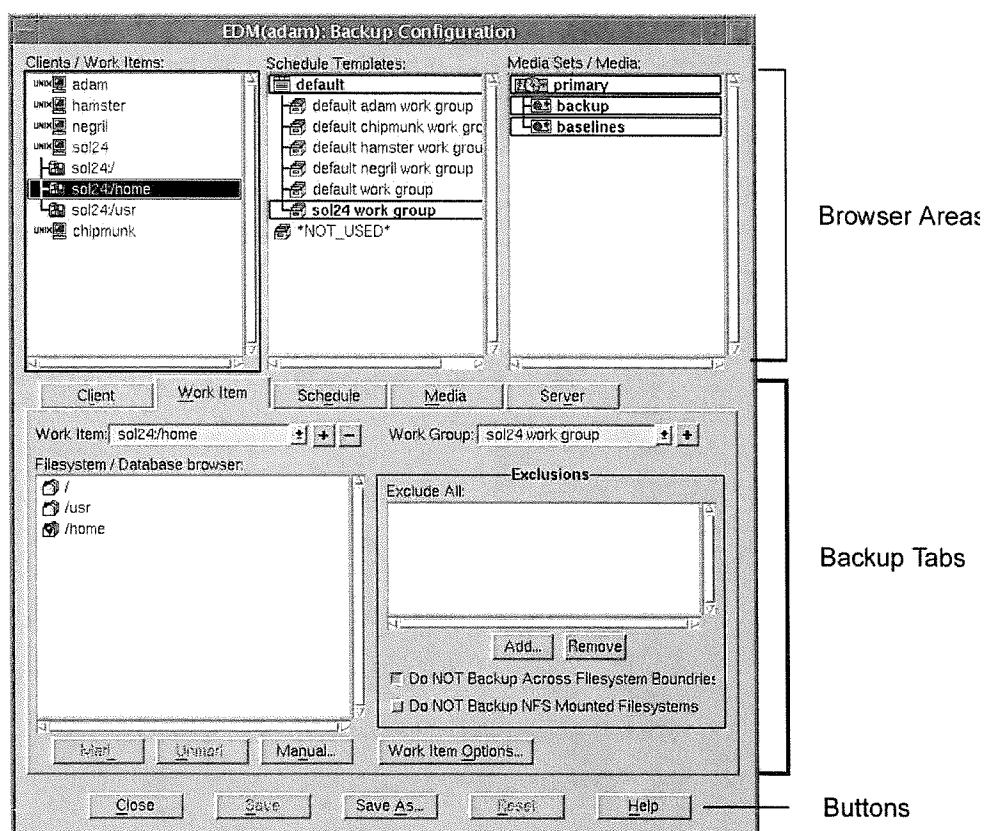
Buttons: Close, Requests..., Duplications..., Help, Print

Backup Configuration



Click on this icon to display the Backup Configuration window. Use it to view backup schedules, configure clients for backups, assign media sets to backups and assign backup administrators to perform restricted tasks.

You can also set backup schedules based on parameters that you specify for your site. This configuration process enables you to tailor the backup system to meet the needs of your organization. For an understanding of backup and restore, review Chapter 3 “Basic Backup and Restore Concepts,” Chapter 5 “How Backup and Restore Work,” and Chapter 6 “Database Backup and Restore.”



Backup Report



Click on this icon to display the Backup Report window, where you can execute reports on active and completed backups. You can view reports on the local EDM, or participants of an EDM domain can view domain reports on other EDMs in the domain.

This window enables you to create, modify, and save backup reports in several key areas, such as performance within specified time periods, work items with poor performance, or failed work items.

You can configure a report that filters on time and date ranges, work item characteristics, backup states, throughput information, and backup level. You can also designate the columns the report displays.

CDM(wombat): Backup Report

File Edit Domain Options Help

Local Reports (wombat)

- Active Backups
- Active and Queued
- Most Recent Backup**
- Queued Backups

Report Run Time: Tuesday, August 31, 1999, 12:44

Work Item	Level	Start Time	Start Date	Running Time	State
wombat/LOCAL_DATABASE	0	08:55:08	09/31/99	00:03:00	Missing or Invalidable configuration file
wombat/home	0	08:55:03	09/31/99	00:03:00	Missing or Invalidable configuration file
wombat/usr	0	08:54:52	09/31/99	00:03:00	Missing or Invalidable configuration file
wombat/var	0	08:54:44	09/31/99	00:03:00	Missing or Invalidable configuration file
wombat/opt	0	18:08:45	09/30/99	12:45:47	Volume or Media error
cable.sbsystem/usr	0	10:00:45	09/30/99	00:02:20	Unknown error occurred
pegasus/usr	3	10:04:29	09/30/99	00:02:49	Successful
pegasus/tmp	3	10:03:20	09/30/99	00:01:01	Successful
stylesus/data1	3	10:03:08	09/30/99	00:01:35	Successful
vigo/usr	0	18:01:33	09/30/99	00:02:21	Unknown error occurred
vigo/ftpusr	11	18:11:45	10/30/99	00:01:27	Unknown error occurred

14 Work Items 18:00:06 09/30/99 12:54:28

Basic Advanced Columns Advanced-Report

Objects: wombat

Date: Entire History Date Range From: 09/01/99 To: 09/30/99 Started in Last: 1999

Time: Time Range From: 00:00 To: 23:59 Occurrences: All Occurrences Most Recent

State: All States Selected States Successful Failed Active Queued

List of Reports

Report on Backups

Summary

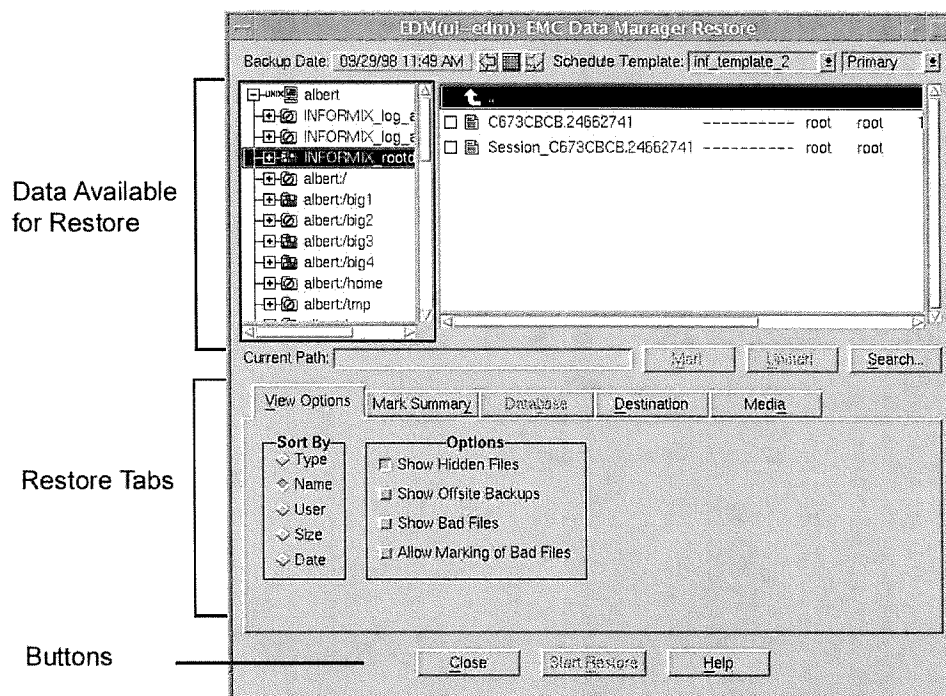
Report Tabs

Restore



Click on this icon, or enter **edmrestore** on the server command line, or enter **edmcrestore** on a client command line to display the Restore window. Use it to restore data that the EDM has backed up.

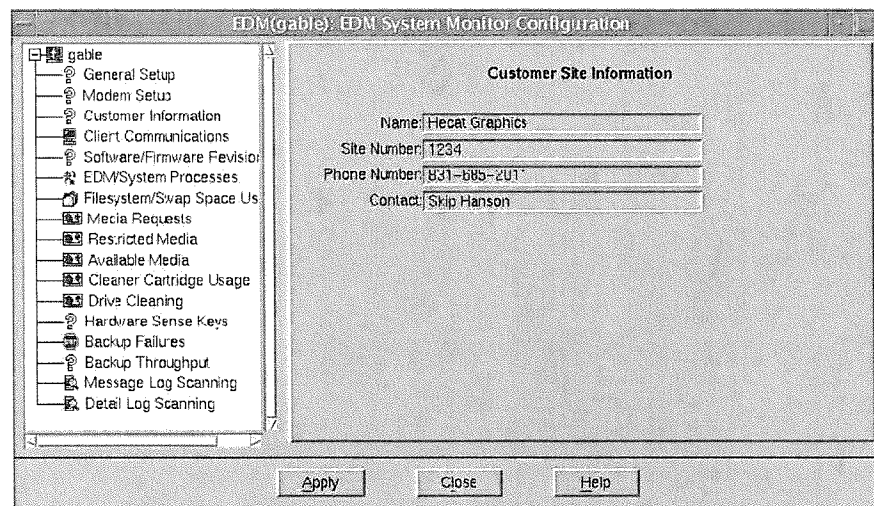
To simplify restoring files, catalogs of backed up files are maintained on disk. Using the restore program, you can browse on-line catalogs, mark files or entire directories for restore, and restore them to a selected client. For an understanding of restore, see “How Restore Works” on page 5-16.



System Monitor Configuration

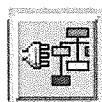
In the GUI, you can configure many RASD and RSM parameters. RASD creates various alerts that can be sent by email to a unique list of recipients, which includes Customer Service personnel and/or the system administrator, an SNMP trap or Tivoli Foundation Level event notification to a designated workstation, or calls home to the EMC Customer Service Database system. The type of alert depends on the severity of the problem, the alert recipients, the length of time that the problem persists. Functions that RASD monitors are scalable and configurable. RASD functionality enables monitoring of items such as available volumes using watermarks, cleaning cartridges using watermarks, drives that need cleaning, unsatisfied volume requests, swap space and filesystems using watermarks, failed backups, client availability, patches and firmware revisions, failed duplications, read/write errors. etc.

The System Monitor Configuration GUI interface is part of the EDM Main window. Under the Main window, choose System Monitor > Configure...

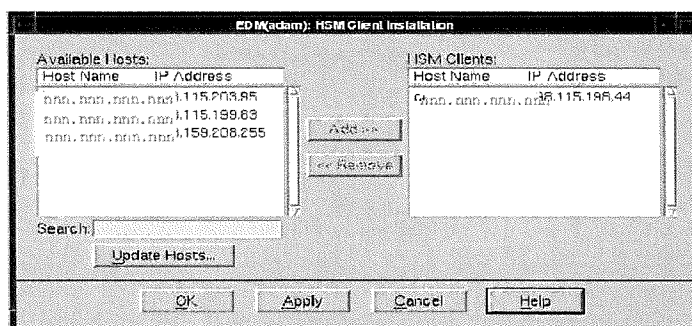


Also part of the EDM RASD software is the ability of Tivoli Foundation Level integration so that Tivoli customers can call the GUI from their framework package. Administrators can launch the EDM GUI from Tivoli, customize their management GUI to call the EDM GUI, provided that their Tivoli Event Console (TEC) is a valid EDM backup client. You can learn more about RASD through the GUI online help.

HSM Client Installation



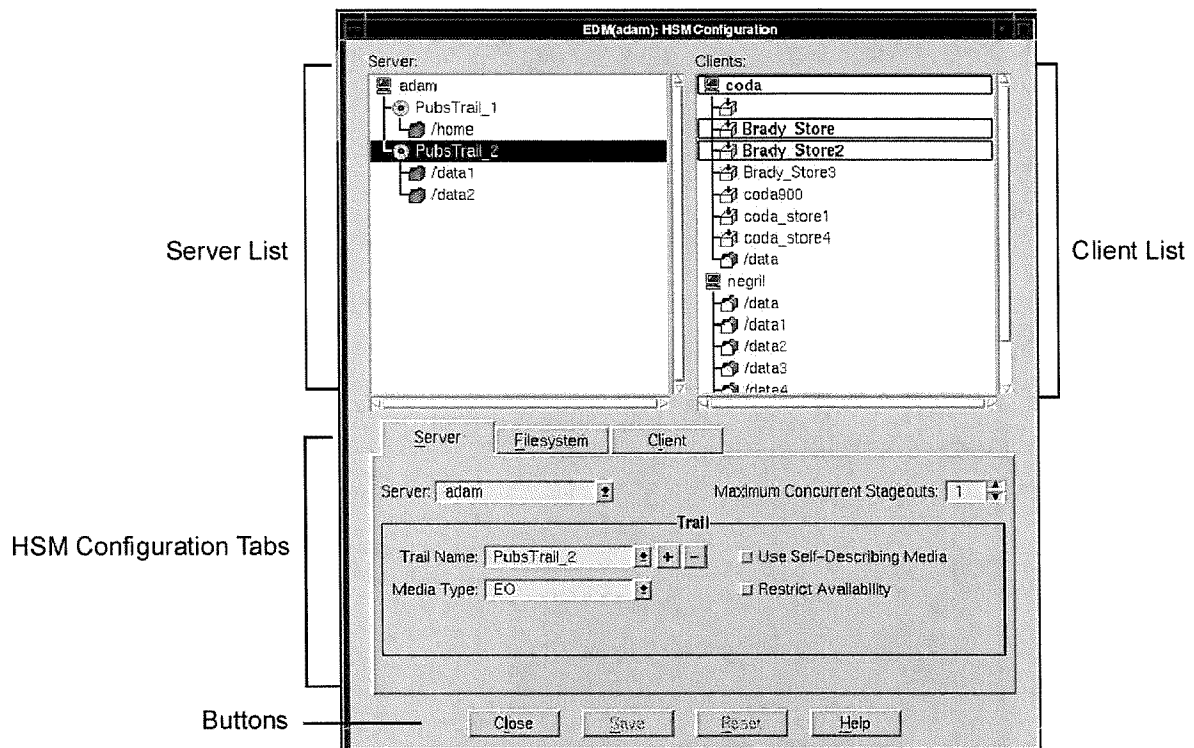
If you have the HSM option, click on this icon to display the HSM Client Installation window to make an HSM client (on which HSM software was previously installed through ep_install) visible and configurable from the HSM Configuration window.



HSM Configuration



If you have the Hierarchical Storage Management (HSM) option, click on this icon to display the HSM Configuration window to set up stageable filesystems on the server and clients. This window enables you to set up the details of migrating file data out to optical disks or magnetic tapes, creating a much larger virtual filesystem. For detailed information about HSM, review Part II, "Hierarchical Storage Management (HSM)," in this manual.



2 Managing Your EDM System

The *EDM Software Reference* provides you, the system administrator, with step-by-step instructions for managing the EMC Data Manager (EDM) system with or without the Hierarchical Storage Management (HSM) option.

This chapter outlines the procedures that you perform regularly to manage your system. Topics in this chapter include:

- Administering the System
- Running Procedures Automatically via Cron
- Backing Up Server Database Files

Administering the System

With an EMC Data Manager system, you should perform several tasks at the beginning of the day.

Most repetitive tasks are set up to run automatically from root's crontab file. This section briefly describes both manual and automatic tasks.

If you have the HSM option, remember that even though HSM software can migrate files to and from staging media, the files must be backed up, whether they reside on the server's magnetic disks or on the staging media. As with any data storage solution, backups are a must.

As a system administrator, you should perform the following tasks on a regular basis:

- ☐ Check daily to ensure an adequate supply of media is available in the library unit(s) for the next backup and migration runs.
- ☐ Verify that at least one cleaning cartridge is available in the library unit.
- ☐ Check the Backup Report window in the EDM graphical user interface (GUI), or the history report, to ensure that your daily backups (and duplications, if any) completed successfully.
- ☐ Read the daily message log file to review system activity.
- ☐ Run and save the backup disaster report.
- ☐ With the HSM Option, check compaction of staging and baseline volumes (weekly).

Database/system administrator duties for Symmetrix Connect backup and restore activities are described in the *EMC Data Manager Symmetrix Connect User Guide*.

Checking Media

Every day you should make sure that enough media is available in the library units to perform scheduled tasks such as the nightly backup and periodic staging runs. Always keep a supply of pre-labeled volumes in the library unit. Also ensure that a cleaning cartridge resides in the library unit. Refer to Chapter 7 “Basic Volume Management Concepts” for more information.

Verifying Backup Completion

You can check on completed backups at any time by viewing the Backup Report window in the EDM GUI. Access this window from the Main window by clicking on this icon in the Main window tool bar.



In the Backup Report window, you can view and execute reports on the local EDM or on a group of EDMs that are clearly defined as a Domain. The report can include backup attributes for a specific work item such as its backup status, total throughput, total size of the backup, total files that were backed up, and others. For detailed information about the Backup Report window, click the Help button in the window.

You can also run **ebreport backup** every day, after the daily backup is scheduled to complete, to verify that all of the scheduled backups completed.

The following example lists all of the backups that have run since a specified date:

```
# ebreport backup -since date
```

The following example lists all of the backups that have run for a particular client since a specified date:

```
# ebreport backup -client clientname -since date
```

Refer to “Backup Duplicate Reports” on page 16-12 and the **ebreport** man page for more information.

Reading the Daily Message File

The error logging facility produces a log file on a daily basis that reports system activity for the past 24 hours. This log file is located in `/var/adm/epoch/daily`. (See Chapter 15 “Message Logging” for more information.) If you have a mail facility, you can edit root’s crontab file to mail this log file to the appropriate users.

The portion of the crontab file that describes the daily log file follows:

```
# Daily log
# eptrunclog truncates /var/adm/epoch/daily to a one day slice.
# eptrunclog can also mail the log to one or more specified users.
# eptrunclog should not be run at the same time as epnewlog
# to mail the log to different user(s), replace "root" with the user list
# to skip mailing the log, delete "root" from the command line
00 7 * * * /usr/epoch/lib/eptrunclog "root" > /dev/null 2>&1
```

By default, the `trunc_daily_log` script mails the log file to root. To send the log file to additional users, do the following:

1. Log in as root.
2. Open root’s crontab file with the editor of your choice.
3. Locate the line:

```
00 7 * * * /usr/epoch/lib/eptrunclog "root" > 2>&1
```

4. Add the names of the users to whom you want to send the log files. Separate user names with a space. For example:

```
00 7 * * * /usr/epoch/lib/eptrunclog "root cbr" > 2>&1
```

5. Save the file and exit from the editor.

Saving the EDM Backup Disaster Report

At the completion of every backup, the `/usr/epoch/EB/config/local_db_cleanup` script automatically generates a MINIMAL Disaster Report. By default, this report is e-mailed to all EDM Backup administrators, appended to `/usr/epoch/EB/config/disaster-report.log`, and printed to the default system printer.

CAUTION: It is essential that, for each backup, you save a hard or soft copy of the MINIMAL Disaster Report in a fireproof location, either offsite or in an onsite fireproof vault.

This MINIMAL Disaster Report is a subset of the FULL Disaster Report generated by **ebreport disaster**. It provides essential information that you need to perform a disaster recovery on the server — a list of media volumes for the most recent LOCAL_DATABASE backup, the current EDM Backup configuration, the current Library Manager configuration, copies of the key configuration-file settings, and information about baseline backups.

This MINIMAL Disaster Report does *not* include backup client information.

You should run the FULL Disaster Report once every backup rotation and whenever significant system changes were made. The following example runs the FULL Disaster Report and redirects it to a file:

```
# ebreport disaster > ~sysadmin/disreports/960917
```

Refer to Chapter 19 for more information on being prepared for a system disaster. See “Backup Disaster Reports” on page 16-19 for a description of the FULL Disaster Report.

Check Compaction of Staging and Baseline Volumes

In the HSM option, compaction is a collection process that, in effect, frees up staging volumes and baseline volumes in a library unit, which ensures a pool of available media.

You can configure HSM software to compact staging volumes automatically. (See the **emcompact** line in root's crontab file.) If you use baseline backup, you need to compact your baseline volumes manually. Note that only reusable media can be compacted automatically. Refer to "Compaction of Staging Media" on page 11-27 and "Compacting Baseline Media" on page 11-29 for details.

Running Procedures Automatically via Cron

You run most repetitive procedures automatically from root's crontab file (refer to on page 2-7). The **cron** entries are placed in root's crontab file during configuration. If you do not use the autoconfiguration option, the configuration programs prompt you for crontab entries.

Note: Adding new entries to the crontab file through the EDM GUI does not replace existing entries with the same characteristics. You must remove existing entries from crontab manually. It is recommended you do this by running **crontab -e** as root. (Refer to the crontab(1) man page for more information.)

You can also schedule a backup in the crontab file within the Backup Configuration window of the EDM GUI. In the Schedule tab, you choose a work group for backup; select Schedule in CRON in the CRON Options section, and enter the time that the backup is to occur. You can also indicate whether you want to retry a failed backup, or use new media for a backup. EDM places this information in the crontab file; the backup then runs automatically at the specified time, on a daily basis.

For maximum efficiency and backup coverage, EDM Backup default settings back up the server first, followed by the network clients, and finally, the EDM Backup and Volume

Management databases. You can back up the server, several clients, and the server database all within a single backup template.

The following entries are added during EDM Backup configuration:

```
# Entries for EpochBackup end with this comment: #EPCebs
# Invoke EpochBackup backup program #EPCebs
00 18 * * * /usr/epoch/EB/bin/ebbackup default >/dev/null 2>&1 #EPCebs
# Invoke EpochBackup catalog cleanup program #EPCebs
00 1 * * * /usr/epoch/EB/bin/ebcatclean >/dev/null 2>&1 #EPCebs
# Invoke EpochBackup catalog expiration program #EPCebs
00 11 * * * /usr/epoch/EB/bin/ebexpire -expire -purge >/dev/null 2>&1
#EPCebs
```

The following table lists procedures that are often run from **cron**. Note that most crontab lines that are used to invoke procedures must explicitly set the full path.

Table 2-1

cron Procedures

Procedure:	Frequency:	For further information:
Run epnewlog to rotate, archive, or truncate EDM system logs.	Hourly and weekly	See “Rotating Error Logs” on page 13-8 and the epnewlog man page.
Run emvck to check and correct staging volume statistics (HSM only).	Daily	Refer to the emvck man page and refer to the daily message log.
Run periodic staging (emmasterd.pid line in root’s crontab (HSM only).	Daily	See “Filesystem Configuration and Maintenance” on page 11-3 for details.
Run emcompact to compact staging volumes automatically (HSM only).	Daily	Refer to “Compaction of Staging Media” on page 11-27 for details.
Run eptrunclog to truncate the daily message log file and optionally, to mail a copy to specified users.	Daily	See the eptrunclog man page.

Table 2-1

cron Procedures (Continued)

Procedure:	Frequency:	For further information:
Run epcleanup to remove files that are no longer needed.	Daily	See the epcleanup man page.
Run ebbackup to back up your server, EDM Backup clients, and EDM Migration clients.	Daily	Refer to “Backup Processing” on page 14-2, and the ebbackup man page.
Run ebreport disaster once every rotation period to keep track of backup media. (A minimal disaster report is generated automatically after each backup.) This report should be stored in a safe place (on another system and on hard copy) for use in recovering from a disaster.	Daily	Refer to “Backup Disaster Reports” on page 16-19 and the ebreport man page.
Run ebreport history after every backup session to see which systems were successfully backed up. This can be included in a short script which also sends mail to the user community.	Daily	Refer to “Backup Duplicate Reports” on page 16-12 and the ebreport man page.
Run ebbackup -drain together with ebbackup -halt if you want to ensure that backup terminates at a certain time.	Daily	See the ebbackup man page.
Run ebcatclean to delete incomplete backup catalogs that may have been created by failed backups.	Monthly	See the ebcatclean man page.
Run ebexpire to manage expiration of catalogs, media, saveset records, and incomplete backups.	Weekly	See the ebexpire man page.
Run ebexpire -purge to delete expired catalogs, media, saveset records, and incomplete backups.	Weekly	See the ebexpire man page.
Run ebexpire -partial to delete incomplete catalogs and backups.	Weekly	See the ebexpire man page.
Run ebexpire -list_orphans to display a list of orphaned volumes.	Varies	See the ebexpire man page.
Run ebexpire -free_orphans to display the list of orphaned volumes and then deallocate them to make them available.	Varies	See the ebexpire man page.

Table 2-1

cron Procedures (Continued)

Procedure:	Frequency:	For further information:
Run emsccheck to clear incomplete bit files from client stores (HSM only).	Daily	See the emsccheck man page.
Run emsundel to recover bit files from the server's backup volumes. (HSM only).	Daily	See the emsundel man page.
Run evmclean to clean tape drives.	Varies	See the evmclean man page. Also, refer to the tape drive's manufacturer for details regarding maintenance scheduling.

Deleting Existing Entries in the crontab File

Adding new entries to the crontab file through the EDM GUI does not replace existing entries for the same activities. You must remove existing entries from crontab manually. After creating new, equivalent entries through the GUI, it is recommended that you delete existing entries by running the command **crontab -e** as root (refer to the **crontab(1)** man page for more information).

Note that the EDM GUI controls pre- and post-commands creation; thus, you do not have direct control over those extensions to the **ebbackup** command when you are using the GUI. The GUI does not allow you to configure complex or non-standard pre- and post- commands. If you want to use complex or non-standard pre- and post- commands, use the **crontab -e** command.

Backing Up Server Database Files

All configuration, backup, and volume information resides in individual server database files. EDM Backup and Volume Management each maintains its own information. The server database files consist of:

- volume management database
- backup catalogs
- backup management files

The LOCAL_DATABASE work item, which is created as part of server autoconfiguration, includes the pathnames of the server database files. It is essential that this work item is part of the nightly backup. The server database files are essential for performing a complete restore of the server and clients in the event of a disk failure.

Always back up the EDM Backup databases on the server after you back up the clients. This is also the case even if you have no network clients, because the server is considered a “local” client to EDM Backup.

Database backups provide you with complete information about both the server and client backups and shorten disaster recovery time because they allow you to restore the database independently from the files that were already backed up. By default, the LOCAL_DATABASE work item is backed up last.

If the LOCAL_DATABASE work item remains in the schedule for more than 24 hours without being run, it is forced to run immediately. This is known as a “late” LOCAL_DATABASE backup.

For more information about the backup and volume management files which make up the server database, refer to Appendix A “Directory Structure”. Backup catalogs are described in “Cataloging of Backup Data” on page 3-11.

3 Basic Backup and Restore Concepts

The EDM Backup software automatically backs up computers throughout your network. It works with volume management software, which manages storage media in robotic library units.

This chapter describes the basic concepts of backup configuration and operation. Its focus is on filesystem backup over the network.

The topics in this chapter include:

- Client/Server Architecture
- Key Processing Concepts
- Configuration Options
- Reports and Logs
- Manual Operations

For information on database backup, see Chapter 6, “Database Backup and Restore”.

For overview information on EDM Symmetrix Connect backup and restore, refer to the *EMC Data Manager Symmetrix Connect User Guide*, the *Oracle Backup Client* guide, and the *EMC Backint* guide.

Client/Server Architecture

Backup and restore software has a client/server architecture:

- *client* software runs on the server (the EDM) and on each client that you want to back up in your network
- *server* software runs centrally on the EDM that also runs the volume management software

Server software automatically administers backups of the data on clients throughout your network and of data on the EDM server itself. It does so according to general parameters shipped with the system and added when you first set up the system. You can change these backup parameters to meet the specific needs of your site.

Remote client software, which is located on each client, receives instructions from the server, scans filesystems, and sends the backup data to the server. Local client software, which is located on the server, backs up the server data to the server's tape library unit.

For level-9 incremental backups, the client software determines which files changed since the last full backup and backs up only those files.

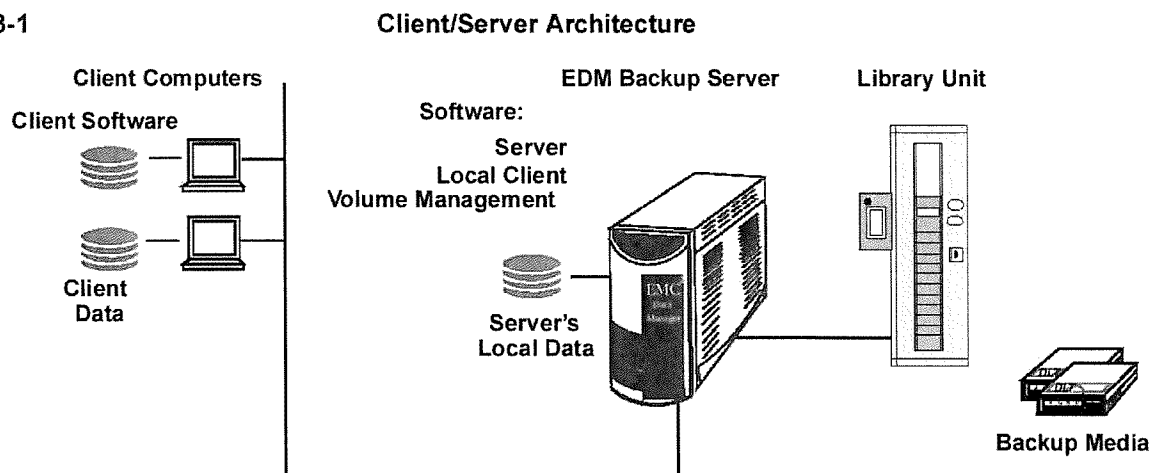
If you purchase the optional EDM Online Database Backup software, you can back up a database without shutting it down.

Without the online option, the basic EDM Backup client software enables offline database backup by shutting down databases prior to backup. After database backups are finished, the client software puts the database back online.

During the restore process, the client software receives the restored data and writes it to the client disk.

For a discussion of Client/Server processing, see "Client/Server Processing Methods" on page 5-6.

Figure 3-1



Key Processing Concepts

EDM Backup scheduling and processing is automatic and dynamic. Workloads are balanced for speedy and efficient backups every night while creating a manageable set of media for every few weeks' worth of backups.

The data you want to back up is specified in *work items*. The data can be a filesystem, disk, or partition on a UNIX, NetWare, OS/2, or Windows NT platform or an Oracle, Sybase, Informix, or SAP R/3 database. (Refer to the *EMC Data Manager Software Release Notes* for a current list of available clients.)

Work items are collected into *work groups*. A *backup schedule template* specifies where to write your backups and how work items are to be scheduled for backup. You can create separate trails for full and incremental backups. Trails are grouped into *trailsets* (media sets), which specify trails for all of the backup levels that the schedule template uses.

This section discusses the following concepts:

- Scheduling
- Nightly Backup Processing
- Restore Processing

Scheduling

EDM Backup offers several ways to schedule backups:

- Automatic Scheduling
- Custom Scheduling
- Command Line Scheduling
- Backup Activity Wizard

If any client is unavailable for backup, EDM Backup continues to back up the other clients in the work group. EDM Backup automatically reschedules failed clients and balances the entire schedule.

Automatic Scheduling

Automatic scheduling of filesystem backups performs some number of full backups each day for the scheduled period. EDM Backup calculates its own schedule for performing full (level 0) and incremental (level 9) backups. With automatic scheduling you can also change the schedule to perform full backups only during weekends. Configure automatic scheduling in the EDM Backup Configuration window.

Note: EDM-initiated database backups are automatically scheduled, too. But with database backups, no determination of full or incremental is made by the EDM backup scheduler. The EDM kicks off the database backups on the database client. The backups are performed according to configuration on the client. See “Database Network Backup Overview” on page 6-7 for more information.

Custom Scheduling

Custom scheduling through the EDM Backup Configuration window enables you to perform filesystem backups other than the levels 0 and 9 backups that automatic scheduling provides. With custom scheduling, you explicitly specify the days and levels of backups for individual clients.

Command Line Scheduling

Command line scheduling enables you to enter command overrides to the schedule template configured by the previous methods. You can use command line scheduling to resume an incomplete backup operation or to manually run a backup that is not currently scheduled with the automatic or custom methods.

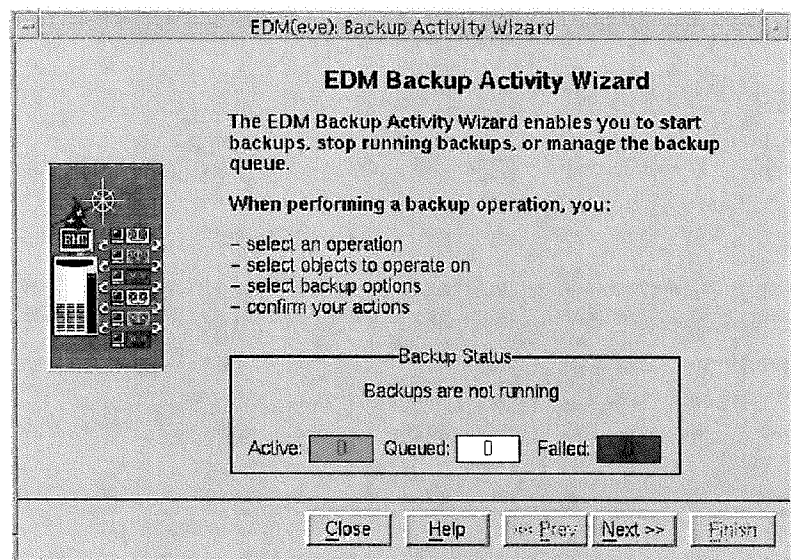
Backup Activity Wizard



Click this icon in the Main window toolbar of the EDM GUI to access the Backup Activity Wizard. This wizard enables you to start new, queued, or failed backups, stop running backups, or manage the backup queue.

Note: You must have root privileges or be an EDM Backup Administrator to use the Backup Activity Wizard.

Refer to EDM online help for more information about the Backup Activity Wizard.



Concurrent Work Item Input

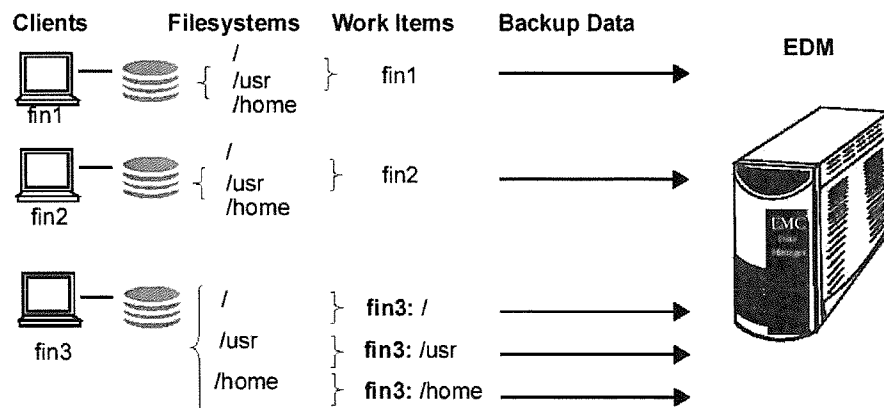
Backup software enables you to gather backup data from many sources at once. The server can concurrently process multiple streams of data from numerous clients.

For a typical client disk, the entire disk, with all its filesystems, is designated as a single backup *work item*. Each work item is backed up by a single client process, sending a single stream of data to its corresponding server process.

Software also considers various limits on backup processing, such as the total amount of backup data in the network, the total concurrent backup streams allowed to be sent to the server at any one time, and the total concurrent backup streams to be funneled together to the storage media.

Figure 3-2

Items of Data Specified For Backup



Large filesystems are designated as separate work items to prevent streams from being too long. (Special parameters prevent conflicting concurrent backups of data from the same disk.)

Balanced Scheduling

Because numerous work items can be scanned for backups separately, an optimal subset of work items can be scheduled for a full backup each night, while the rest of the work items receive incremental backups. Server software intelligently schedules cyclical full backups along with nightly incrementals. This is known as *autoscheduling*.

If you choose to autoconfigure your system, you have the following configuration:

- Every client is backed up according to a single backup schedule template. The backup template lists the default work group.
- Work items are created for each client and inserted into the default work group.
- Backups are scheduled automatically, with each client receiving at least one full backup every two weeks (the rotation period) and receiving an incremental backup on all other nights.
- Backups are written to one media set (trailset) every night.
- All of the data that is sent to the trailset is written to a single media volume or series of volumes (a single trail). (A media volume is a labeled tape cartridge or one side of an erasable optical disk.)

Rotation Period

Software rotates full backups among the work items so that all work items receive a full backup at least once within a *rotation period* (the default is 14 days).

Load Balancing

Every work item is scheduled for either a full or incremental backup each night. To create the nightly backup schedule, the server software considers not only the rotation period, but also actual backup results from previous nights. If all work items received a full backup in the rotation period, the software's *load*

balancing feature will schedule a new full backup for some work items, to continue to smooth out the backup work load for each night.

Multiplexed Storage

While numerous work items can be scanned for backups separately, it is possible to multiplex (funnel) the backup data together when writing to the storage media.

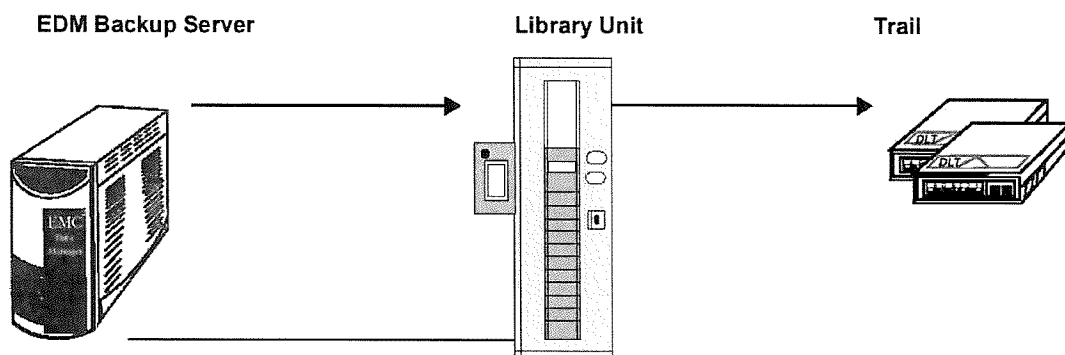
Trails

By default, the server software sends all of the backup data in a single stream of data to a single backup drive, where the data is written to the backup volume. Whenever the volume fills, a new volume is automatically inserted into the drive. The next night, another single stream of data is again written to the volume. At the start of the next rotation period, a new volume is automatically inserted in the drive.

The single media volume or serial set of media volumes written to over the course of one rotation period is called a *trail*.

Figure 3-3

Single Trail

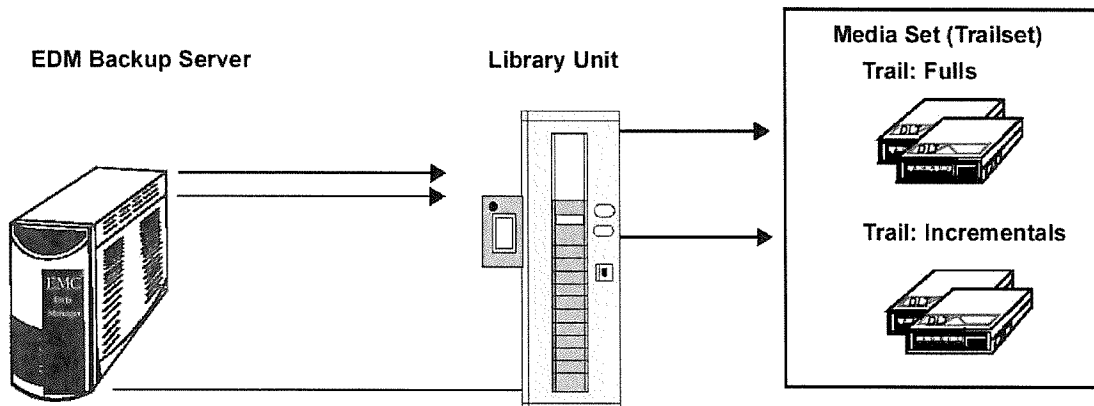


Trailsets

A complete set of full and incremental backups for a rotation period is called a media set (trailset).

By default, the single trail also constitutes the trailset. But, you could write full and incrementals in two separate streams to two separate trails (sets of media). You would need to use two drives for writing the backups concurrently.

The combined set of full and incremental trails constitutes the trailset. Each trailset contains at least one full backup for each work item (from various nights in the rotation period) plus the incremental backups for every other night in the rotation period.

Figure 3-4**Multiple Trails in a Trailset**

Nightly Backup Processing

Server software runs automatically every night to administer network-wide backups. The software includes configuration parameters describing what data gets backed up from the magnetic disks of the server and any client computers. The software also maintains status information about what data was or was not backed up recently.

Software refers to configurable *backup shift* guidelines for the maximum number of hours that the entire nightly backup has to start backups. One guideline applies to weekdays, another to weekend shifts.

Client software scans the filesystems and databases (both filesystems and raw partitions) on the hosts and streams the data to the server.

The server writes the data to tape library units attached to it through SCSI connections.

Server software creates online catalogs of the backups to make it easy to restore whenever necessary. Notifications by email inform you of the status of your backups. You can run reports as well.

Automatic Start

A main backup process is started nightly by **cron** from the root crontab file. (For more information, see “Running Procedures Automatically via Cron” on page 2-6.) This process consults the configuration parameters and status information, and then automatically schedules backups for each client’s magnetic disks.

The main process spawns separate processes to handle the backup work for each client or portion of the client’s data. The individual server processes each send instructions to their corresponding client, and specifies which filesystems and files to back up.

Client Processing

On each client computer (and on the server computer itself) the client software scans the local filesystems as directed.

When prompted for a *full backup* (level 0), the client software copies data for all filesystems, directories, files, and databases specified for backup and streams the data to the server.

When prompted for an *incremental backup* (level 9) the client software sends file data for only those files that have changed since the previous backup. (Unlike the UNIX **dump** level 9, each consecutive level 9 backup copies only files that have changed since the last level 9 backup.)

Whether it is doing a full or incremental backup, the client software always sends complete directory information, so that during the restore process, you can browse an accurate view of the directories as they were at the time of any backup.

Backups of Changing Files

You can continue to work during backups. As the backups are processing, the server software checks the backup directory information to find files that have changed during the backup. Any such files are backed up and checked again two more times that session. (Any files that continue to change both times will be backed up during the following backup session.)

Storage of Backup Data

As the server software receives the backup data, it writes all the data (from multiple clients) to one or more drives in the Library Unit. By default, it streams all of the data to a single backup drive, writing the backup data to the volume in that drive.

Cataloging of Backup Data

Backup software creates *catalogs* that keep track of filenames, file attributes, and locations of backup data. It copies the catalogs onto the media along with the backup data. When a backup completes, the software processes the catalogs and keeps them online so that you can quickly retrieve the data.

To maintain sufficient magnetic disk space for backup catalogs, you'll need to expire the older catalogs after a fixed length of time. The catalog expiration period must be at least one day longer than two rotation periods. To determine catalog expiration, you must consider other system factors and user requirements for executing restores. Catalogs have dependencies on backups and other catalogs. Before you decide on an expiration schedule, refer to "Expiration of Backups and Catalogs" on page 10-2 for more information.

Restore Processing

Anytime you need to restore a file from the backup media, you or your users can use the **edmcrestore** command on the client to open the Restore window in the EDM GUI and display it on the client to browse and restore backed up data. Of course, you can also open the EDM Restore window directly from the EDM Main window on the server.

The Restore window displays file listings derived from the online catalogs. You can browse through directories as they existed on each backup date, and you can browse back and forth among backup dates.

When you have selected the files you want, you start the restore. The backup software locates all data, restores it to the client, and logs the restore activity to log files on the server and client computers.

Configuration Options

As system administrator, you can configure backups for both the server and clients centrally from the EDM Backup server by using the Backup Configuration Wizard and the Backup Configuration window in the EDM GUI. You can restore the backed up files using the EDM Restore window.

You can configure various aspects of your backups to meet your site's needs including your network's computers, library units and drives, data, user profiles, and workshift scheduling demands. Use the EDM Backup Configuration Wizard to set the parameters within the server's backup *configuration file* (eb.cfg). Use the Backup Configuration window to customize those settings, if necessary. For a description of the fields in the eb.cfg file, see Appendix B "EDM Backup Configuration File".

Key Configuration Options

Adjusting the configuration is, for the most part, optional. The configuration file is shipped with standard default values that are ready to run and are generally suitable for most sites.

Your only required configuration task is to install clients. In essence, all you need to do is to specify on the EDM server:

Which client computers should be backed up?

With clients installed, you can optionally configure these key aspects of how automated backup processing proceeds:

- *What* data on each client do you want to back up?
- *When* should backups run?
- *Where* is backup data written?

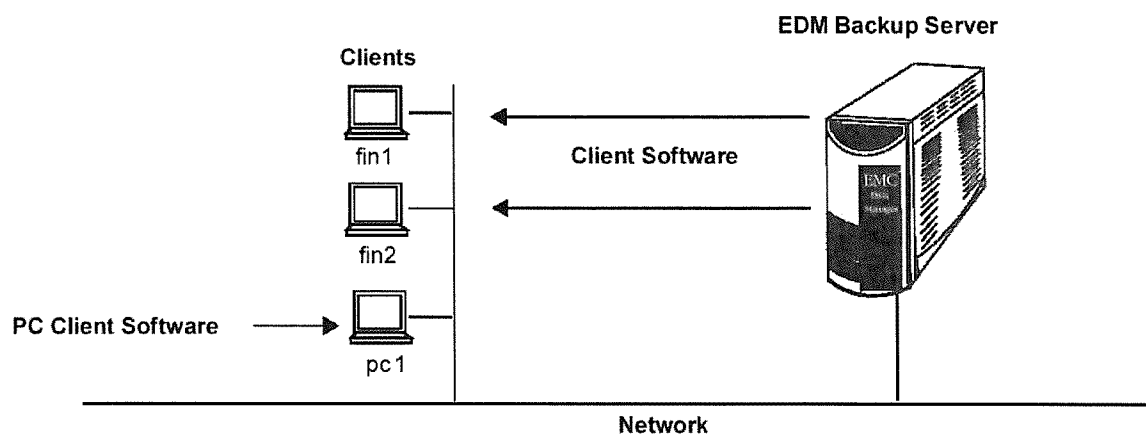
Which Clients to Back Up?

Use the EDM Backup Client Install Wizard on the server to specify which client computers to back up. The Wizard leads you through a process where you select the clients from a list and install the software from the server over the network to the clients, as shown in Figure 3-5.

Note: For PC clients you must first install the client software directly on the PC system as described in the appropriate user guide. Then use the Install Wizard, so that the EDM server recognizes the client.

Figure 3-5

Installing Client Software



Installation Options

The EDM Backup Client Install lets you change various installation defaults before you actually install new clients. This is especially helpful for distinctive client platforms.

Installation options include changing accounts and directories for the client software and communication timeouts.

What Data is Backed Up on Each Client?

The specification of backup work for each client consists of one or more work items. The work item description or *directive* specifies filesystems, directories, files, and databases that are included and excluded from backup. The work item directive is an expanded version of a UNIX **find** statement called a **findxcpio** statement. For more information, see Appendix D "findxcpio Directives".

Use the Backup Configuration Wizard to specify the data you want to back up. Then you can use the Backup Configuration window to edit single work items.

When are Backups Scheduled?

The general parameters for backup scheduling are handled within a *backup schedule template*. The schedule template provides parameters that the autoscheduling function uses. It ties them together with the work items to be backed up and the media set (trailset) to be written to. Use the Backup Configuration Wizard to set these values.

Rotation Period Trade-offs

The rotation period and the various other scheduling parameters are specified in the schedule template. The rotation period specifies a value that is actually used for two purposes:

- Schedule rotation
- Media rotation

The schedule rotation is the fact that the software must perform a full backup on each filesystem work item covered by the template during this period.

The media rotation is the fact that the software appends each nightly backup to the current volume during this period; at the start of a new rotation period, a new volume is started.

The most common rotation periods are 7 days, 14 days (the default), or 28 days. A key consideration in determining the rotation period is that it is faster to write incremental backups than full backups, and incremental backups use up much less

storage media. But because it is faster to restore files from a full backup than from a series of individual incremental backups, you might want to specify shorter rotation periods for data that changes frequently. If your data changes less frequently, you might want to specify a longer rotation period.

Assigned Work Groups

Work items are grouped into one or more *work groups*, whose backups are coordinated by one or more backup schedule templates.

By default, all server work items are assigned to a single work group named “default server work group.” All additional client work items are assigned to a single work group named “default work group,” and both work groups are assigned to a single schedule template named “default.”

Use of Additional Backup Schedule Templates

All client backup work can be coordinated by one schedule template, but if some client backups need to occur more or less often you can assign some of the work to a separate template and specify different rotation periods for each template. This will write the data for groups of clients to separate media.

Where is Backup Data Written?

The primary reason for choosing a certain scheduling option may have more to do with the resulting use of media. For example, multiple work groups, schedule templates, and media sets (trailsets) are useful for organizing a site for accounting purposes; by separating the media to which the data is written, you can charge each group for the media that the group uses.

The schedule template names its corresponding trailset. By default, all of the data is written to a single trailset. Therefore, you can create completely separate sets of backups for work groups by creating separate schedule templates and trailsets for different work groups.

And, as is mentioned earlier, you can create separate trails for full and incremental backups without editing the schedule template; you just edit the trailset to create a second named trail to receive the data from your full backups.

Note: Moving backup media offsite before its rotation period ends causes the backup that would use that media to fail. You can avoid a failed backup by using new media for the next backup. You configure the use of new media in the Backup Configuration window of the EDM GUI. Select Advanced Options in the Schedule Tab. In the Schedule Options window that appears, select Use New Media When Current (backup media) Is Offsite.

Alternate Trailsets

You can schedule two complete sets of backups on alternate nights and store one set of media offsite. This doubles the number of full backups; one full backup is written to each trailset.

In the EDM Backup Configuration window, you can create two complete sets of full and incremental backups that are written to on alternate nights. To configure two alternate sets of backups, you use a single schedule template, but in that template, you specify an *alternate* trailset in addition to the *primary* one.

The template writes to the primary trailset the first night of the rotation period and to the alternate trailset the next night, and so on. Twice as many full backups are run. In addition, there is more incremental data, because, instead of backing up files that changed in the past day, each incremental backs up files that changed in the past *two* days — since the previous backup *to the same trailset*.

Custom Schedules

In the Backup Configuration window of the EDM GUI, you can create separate trails for full explicit incremental backup levels (1-8). You create a custom schedule within a template and edit the trailset to create various named trails for one or more levels from 1-8.

In a schedule template, you can override autoscheduling and custom-schedule backups of particular levels on particular days for certain work groups. This should be done as an exception rather than the rule, as you are bypassing the software's autoscheduling intelligence and its benefits.

But you can do this just for special data so that you can be certain that it gets backed up on a set date, so that you can specify one or more *explicit* incremental backup levels (levels 1-8) as well as levels 0 and 9, and so that you can write that data to a separate trail.

Media Duplication

Media Duplication allows you to create a duplicate set of backup media automatically after each backup session.

After you configure media duplication in the EDM Backup Configuration Wizard, the duplication of a set of backup media occurs automatically after each backup session. This background activity starts after nightly backups complete.

For a complete description, see Chapter 9 "Media Duplication."

Other Configuration Options

With automatic backups configured, you can also use the EDM Backup Configuration window to configure optionally the following other aspects of the backup and restore software:

- processing of concurrent backups
- identifying individuals who can configure automatic backups, run backups manually, and run restores
- determining the period of time that backup catalogs and data are kept before expiration
- using new media when current backup media is offsite

How are Backups Processed?

More than one backup process can run concurrently on the server and client and be written to one or more backup drives. When extensively tailoring your configuration, you might decide to tune the preset limits on concurrent processing on the server, clients, and to the backup drives.

You can limit these factors affecting backup and network performance:

- maximum number of work items the server can back up at the same time
- maximum number of work items each client can back up at the same time
- maximum number of work items to concurrently write to all trails for each media type
- maximum number of work items to concurrently write to each particular trail (overridden if maximum for that media type has been reached)
- maximum amount of backup data in the network

Permissions and User Modes

You can configure backup to recognize usernames for accounts on your network as a backup administrator or a various class of restore users. (Keep the usernames to eight bytes or less.)

Backup Administrator

You can configure EDM Backup to recognize your username as a backup administrator, so that you can run the EDM Backup Configuration window. This also enables you to run backups (**ebbackup**) manually from the server under your username.

Administrator Restores

As backup administrator, you have permission to run restore with full permissions as system administrator from the server or any client computer. You can use the EDM Restore window to browse the backups of any client and change the destination client for the restore.

Client User Restore Modes

For each client, you can authorize users to use restore with various levels of permissions.

You can configure these in the EDM Backup Configuration window:

- self-service restore users (who only can restore their own files on a single client).
- cross-client restore users (who can restore their own files over to another specified client).
- root-permission restore users (who can restore any files for a single client).

Catalogs and Backup Savesets

The EDM Backup software creates one *backup saveset* for each work item every time it backs up the work item.

The following comprises each backup saveset:

- *backup data*: a copy of each client's backup data on the storage media.
- *backup catalog*: an online listing of the names and attributes of each directory and file in the work item at the time of the backup and the location of backup data for each file that was backed up. Backup processes this catalog after the backup and uses it when needed to restore data.
- *saveset records*: contain information about an entire backup; for example, its start time and the trails that the backup program uses to write the backup data.

Expiration policies for backup data, online catalogs, and online saveset records are defined for the various backup levels. By default, full backups are kept for one year and incremental backups for three months.

You can configure how long to keep backup data (on the backup media) before expiring it so that the media can be reused.

You can also configure earlier expiration of the online catalogs that reference the backup media; you would do this to maintain magnetic disk space on your server, while keeping backup data longer, just in case it is needed. Catalogs take up a lot of magnetic disk space. See Chapter 10 "Magnetic Disk Concepts" for more information.

The backup software needs the saveset records for as long as you keep the backup data. Saveset records are small relative to the catalogs. With the saveset records online, you can use the **ebimport** command if it is necessary to recreate an online catalog for a backup.

Reports and Logs

Backup software maintains logs of all activity, mails notifications about backup processing, and provides various reports to aid you in monitoring the status of backups.

Backup software maintains logs for backups and restores on both the backup server and on each client. It also maintains, on the server, a log that details the backup activities and cataloging operations for each backup template.

EDM Backup automatically sends email notifications about backups that are in progress, that have completed, or failed. You can modify the management of mail notifications and log files in the Backup Configuration window of the EDM GUI.

Refer to Chapter 15 "Message Logging" and Chapter 16 "Backup Reports and Log Files" for details.

Reporting in the EDM GUI

You can monitor active backup processes and execute reports in the EDM GUI.

During a backup, an object in the Main window such as the EDM server, a client, or a work item appears as an active process, successfully backed up, in the backup queue, or failed to complete successfully. Current backup throughput also appears for a backup in progress.

Upon completion of a backup, you can then configure, save, and print backup reports on specific areas of importance such as failed work items or work items with poor performance.



Click on this icon in the Main window toolbar to access the backup report module.

For more information about active backup reporting, refer to EDM online help, “Backup Report Overview.”

Manual Operations

Much of the backup operation runs automatically, but you can use certain manual operational and reporting commands at the command line or set them to run automatically from the root crontab file. Refer to Chapter 18 for a list of man pages that are available for backup and restore.

4 Port Control

EDM port control allows the EDM to communicate with clients on the other side of a firewall. Port control is available for use with UNIX and NT filesystem backups and with UNIX database backups.

This chapter contains the following sections:

- Understanding Port Control
- What is a Firewall?
- Port Control Checklist
- Setting Up the Firewall for Port Control
- Enabling Port Control on the EDM
- Installing Port Control on the EDM Client(s)
- Making Changes to Port Control

Port control allows you to control the TCP ports used by the EDM to communicate with the clients. It also makes network analysis and auditing of EDM network activity easier. It also allows you to take advantage of the router's ability to prioritize packets.

The discussion in this chapter is limited to how port control allows EDM TCP port usage to behave in a predictable manner so that firewall rules can be implemented. A firewall restricts access between networks based on rules set by the local firewall administrator.

Overview

EDM port control allows EDM TCP port usage to behave in a predictable manner so that firewall rules may be written to allow the EDM to communicate with EDM clients on the other side of the firewall.

The port control feature is available for use with UNIX and NT filesystem backups and with UNIX database backups. It gives you the option to control the TCP ports used by the EDM. Port control functionality allows you to control TCP port usage so that you can:

- Back up and restore UNIX and NT filesystems and UNIX databases in a firewall environment
- Analyze and audit of EDM network activity
- Take advantage of the router's ability to prioritize packets across the network, based on your own requirements

EDM port control addresses only a portion of a complete security solution. When properly configured, it eliminates EDM's dependency upon the backup clients' portmapper. Port control must be coordinated between an EDM and all clients that are expected to use it.

Understanding Port Control

By default, port control is disabled on the EDM. To enable port control on the EDM server use **eb_server_config** without the **-D** option. You can configure an EDM to have some clients using port control and some clients not using port control. To enable port control for selected client(s) use the Backup Configuration Wizard in the GUI, or the **eb_client_install** command with the **-portcontrol** option.

Enabling port control is an easy procedure when implemented correctly at the time of server installation or update. Discuss your firewall policies with the EMC service personnel who install your EDM. To adjust the settings after installation, you must use the **portservices** CLI to change port values, rerun **eb_server.config**, and reinstall all participating clients (see the **portservices** man page.). Contact Customer Service for assistance.

Installing and Updating Client(s)

Installing, updating, and operating an EDM port control enabled client through a firewall has the same network requirements as other EDM clients, such as network name resolution. Port control allows normal operations to take place in definable tcp port ranges.

Installing and updating a UNIX EDM client requires either UNIX rsh or UNIX rexec accessibility from the EDM to the client and the ability to ping the EDM from the client. These protocols are not usually permitted by firewalls and will need to be allowed during the installation or update.

Restrictions

Please note the following restrictions (see Table 4-1 on page 4-6 for default port definitions):

- Port control is not a backward-compatible feature, therefore it requires EDM clients to be updated to EDM 4.5 versions. It supports most versions of UNIX and NT clients (but not Pyramid, Sequent, NCR, SCO and Alpha NT).
- Network database backups require a return connection through the firewall from the client.
- In client-initiated backups, ports have to be open from the client to the EDM.
- EMC does not recommend by-passing firewall policies by bridging the DMZ and trusted network with the Symmetrix. Therefore, if you want to use port control with EDM Symmetrix Path or EDM Symmetrix Connect, you should discuss this carefully with EDM customer service and the local firewall administrator before implementing.

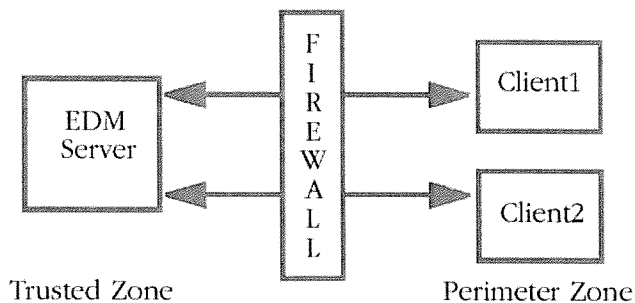
What is a Firewall?

A firewall is a combination of hardware and software applications used to create a gateway and provide controlled access from one network to another. By reviewing all traffic and selectively allowing or disallowing data to pass, the firewall protects the internal network from unwanted external intrusions.

A firewall enforces network policy regarding the restriction of access based upon rules set by the local firewall administrator.

EDM Firewall Assumptions

EDM port control assumes that the Perimeter Network Zone is separated from the EDM by an internal firewall and that it is reasonably secure. If the EDM is in the trusted zone, port control allows it to work within the firewall rules such that the EDM can perform a backup of a client in the Perimeter Zone. Port control can also be set up so that the clients in the perimeter zone can initiate backups.



For normal backup and restore operations, the firewall between the trusted zone and the perimeter zone must be opened to allow TCP communication. The scope of the opening depends on your company policies.

The following are baseline assumptions needed for using EDM port control to backup and restore UNIX and NT filesystems and UNIX databases through a firewall:

EDM port control:

- Assumes an IP filtering-capable firewall (which most firewalls are).
- Uses the TCP protocol. (Use of UDP is not required and ICMP/SNMP are used by RASD for non critical functions.)
- Exists so that rules can be written for a firewall that allow the EDM to interact with EDM clients via TCP. Include the local firewall administrator in the process of defining ports.
- Allows rules to be created within routers to prioritize packets.

Table 4-1

Default Port and Port Range Definitions

	From Server to Client	From Client to Server
UNIX Filesystem Backup and Restore	8000:8250	—
UNIX Database Backup and Restore	8000:8250	5600
UNIX Client Initiated Filesystem Restore	8000:8250	8000:8250
with GUI Display	X11 ports (6000:6063)	—
UNIX Client Initiated Database Backup and Restore	8000:8250	5600
Windows NT Filesystem Backup and Restore	3895	—

- If you are using UNIX portmapper, you need to open tcp port 111 along with 8000:8250.

- Installing and updating EDM clients though a firewall requires the firewall to be open temporarily. These ports may then be closed after installation/update of EDM client(s).

Some firewalls understand these protocols and allow them to be generically specified to take care of the details.

For those that do not, see Table 4-2:

Table 4-2

TCP Connections for Client Installation and Update

To Allow:	From The EDM to the Client	From the Client to the EDM
UNIX RSH from the EDM to the client	514	TCP connections on privileged ports less than 1024
UNIX REXEC from the EDM to the client	512	TCP connections on non- privileged ports greater than 1024, but usually greater than 5K, up to 65K
ping from the client to the EDM		ICMP

Note that the UNIX REXEC protocol passes the password, in this case the root password for the client, unencrypted. For this reason, the local firewall administrator might choose to allow the UNIX RSH protocol and temporarily place a .rhosts file in the root directory on the UNIX client.

Firewall Requirements

Firewalls control connections. Once connections are established, data can flow bi-directionally. IP firewalls need to know about protocol, source address/port and destination address/port. The side of the firewall the connection request originates from (DMZ or Internal Network) impacts required ports.

Firewalls execute rules in order, looking for the first match. Some typical rules are:

- Allow tcp source <edm> any destination <client> 8000:8250
- Allow/drop/reject matching request.
 - allow - like router, forward packets.
 - drop - silently drop packets (*timeout*).
 - reject - notify originator (*connection refused*).
- Protocols are usually TCP, UDP, ICMP.
- Source (from) what address range/port range.
- Destination (to) what address range/port range.
- Source and Destination are specified as universal addresses:
<system name or ip>:<port range>

Examples:

myedm.customer.com:8000-8250

193.45.5.25:8000-8250

193.45.5.0:8000-8250

193.45.5.24:6000

Sample Firewall Configurations

The following examples are provided for the local firewall administrator. These examples assume that the EDM is 123.456.78.155 and the EDM client is 123.456.78.170.

Basic Port Range Example

To accept TCP connections from the EDM to a client within the defined port range:

```
allow tcp source 123.456.78.155 any destination 123.456.78.170 8000:8250
```

This allows any TCP port on the EDM (123.456.78.155) to connect to TCP ports 8000 through 8250 on the client (123.456.78.170).

NT Client Example

To back up a Windows NT filesystem client, you must allow port 3895 on the NT client to accept TCP connections from the EDM:

```
allow tcp source 123.456.78.155 any destination 123.456.78.170 3895
```

This allows any TCP port on the EDM (123.456.78.155) to connect to TCP port 3895 on the client (123.456.78.170).

Database Example

To back up a UNIX database client, you must allow port 5600 on the EDM to accept TCP connections from the client:

```
allow tcp source 123.456.78.170 any destination 123.456.78.155 5600
```

This allows any TCP port on the client (123.456.78.170) to connect to port 5600 on the EDM (123.456.78.155).

Port Control Checklist

Before you begin to enable and configure port control, make the decisions in the Port Control Checklist. The local firewall administrator should participate in this process. These decisions will be used to construct firewall rules and to configure the EDM and participating clients.

It is important to do it correctly the first time.

To change it later will require the use of the **portservices** command line to make changes to the EDM server and then add the changes to the client(s) in order to keep them synchronized. (See “Making Changes to Port Control” on page 4-20.)

Table 4-3

Port Control Checklist

Decisions to Make	Record Decision and Needed Action
<input type="checkbox"/> Name of EDM for port control. (Must be at least EDM 4.5.0)	
<input type="checkbox"/> Decide which EDM clients and/or subnets will be accessed through the firewall. (Must be versions released with EDM 4.5.0 or greater.)	
<input type="checkbox"/> Decide if the default port range 8000:8250 is appropriate or if another port range is preferred.	
<input type="checkbox"/> Decide if you want to use client-initiated backups for any client. If you do, you must be prepared to open the port range from the EDM client in the DMZ to the EDM server. See Table 4-1 on page 4-6 for details.	
<input type="checkbox"/> Note that a low TCP session timeout value could result in failure of mover-aware backups and possibly restores. Either increase the timeout value or only do non-mover backups through the firewall.	

Table 4-3

Port Control Checklist (Continued)

Decisions to Make	Record Decision and Needed Action
<input type="checkbox"/> Decide if you will be using the UNIX portmapper on the client (the default) or the EDM portservices file. <ul style="list-style-type: none"> – Using the UNIX portmapper is conservative, but less secure. This is the default, and requires that TCP port 111 must be open through the firewall. – We recommend using the EDM portservices file. This is more secure since you no longer open port 111, but get the port numbers from local files on the server and the client. (There is a risk that services might not be able to be contacted if the configuration becomes unsynchronized by making changes on only one side). 	
<input type="checkbox"/> Decide if UNIX database backups will be performed. If so, the firewall must allow port 5600 to be open from the client in the DMZ to the EDM in the trusted zone. See Table 4-1 on page 4-6 for details.	
<input type="checkbox"/> Decide if NT filesystem backups will be performed. If so, port 3895 must be open from the EDM to the NT client in addition to the port range. See Table 4-1 on page 4-6 for details.	
<input type="checkbox"/> Decide how you want to install UNIX client(s). Select either EDM Transfer Protocol or Remote Shell. Determine TCP port settings. See “TCP Connections for Client Installation and Update” on page 4-7, for more information.	
<input type="checkbox"/> Decide if Symmetrix Path or Symmetrix Connect backups will be performed. EMC does not recommend by-passing firewall policies by bridging the DMZ and trusted network with the Symmetrix. Therefore, if you want to use port control with EDM Symmetrix Path or EDM Symmetrix Connect, you should discuss this carefully with EDM customer service and the local firewall administrator before implementing.	

Some optional features may not work if UDP and ICMP protocols are not allowed on the firewall, such as RASD client pings and SNMP alerts.

Note: There are 220 reusable ports per client. Database backup work items require 3 ports each plus 1 for each stream. Filesystem backup work items on the client use 5 ports per work item.

After you complete this checklist, proceed with the following:

- Setting Up the Firewall for Port Control
- Enabling Port Control on the EDM
- Installing Port Control on the EDM Client(s)

Setting Up the Firewall for Port Control

After you understand the issues involved in using port control with your EDM, and have completed the “Port Control Checklist” on page 4-10, have the local firewall administrator make all of the firewall adjustments before you configure the server to enable port control.

Enabling Port Control on the EDM

You must configure the EDM server before you install and configure EDM clients. At that time, you will push the definition files out to the client(s) to enable port control.

While it is possible to make some changes later, it is much easier to activate port control when you configure the EDM for the first time. See “Making Changes to Port Control” on page 4-20 for several examples of changes. Therefore, be sure to complete the “Port Control Checklist” on page 4-10 before configuring the EDM.

To configure the EDM, run **eb_server_config** without the **-D** option. You must answer **yes** to activate port control. You only need to do this once. Port control will remain enabled when you run **eb_server_config** again.

After logging in as root on the EDM server, enter:

```
# eb_server_config
.
.
.
Do you wish to enable port control on the server? <y/n>[ n] : y

Port Control Information:

Low Port:          8000
High Port:         8250
Lookup Method:     "portmapper"

Enter the low port number for port control:

    (or just press return to accept the default value in square brackets)

[ 8000] :

Enter the high port number for port control:

    (or just press return to accept the default value in square brackets)

[ 8250] :

You have the following choices for Lookup Methods:

    1) Portmapper
    2) EDM port services file

    (or just press return to accept the default value in square brackets)

[ 1] : 2

Port Control files successfully installed on the EDM server.
```

At this point, port control is enabled on the EDM server and you can enable port control on EDM client(s) when you install them.

Portservices Files

Portservices files are created in the format `edm_services.xxxx` by the **portservices** command in the server's `/usr/epoch/etc/csc` directory. The `csc` directory remains empty until port control is enabled. (See the `portservices` man page for details.)

The files specify which ports an EDM server uses to communicate with its clients and other EDM servers. The presence of these files indicates that port control is configured.

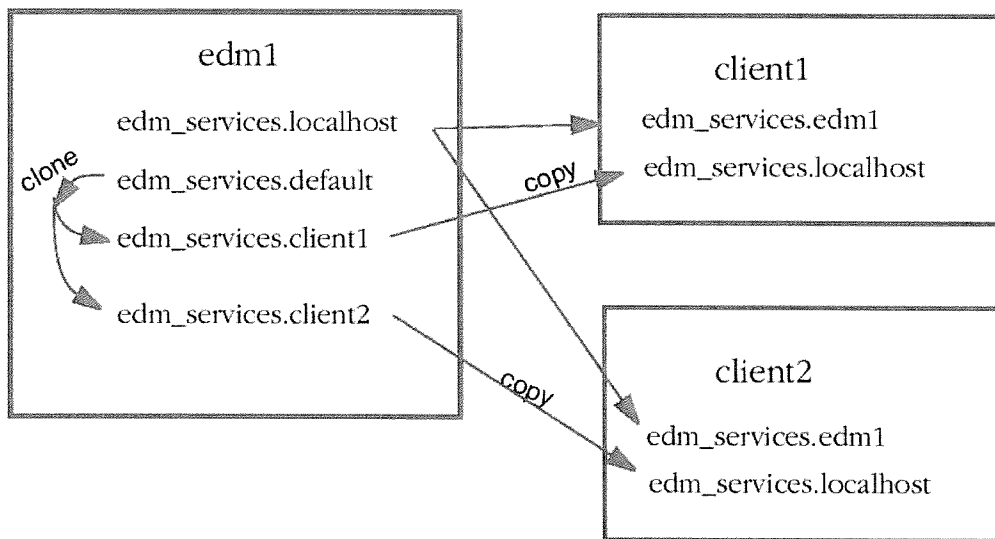
The following examples assume that:

- `edm1` is the EDM server and the local client.
- `client1` and `client2` are EDM clients.
- `.client1_template` contains uncommitted changes for `client1`.

```
-rw-r--r--  1 root    other      816 Jan  7 16:40 edm_services.client1
-rw-rw-rw-  1 root    root       824 Jan  9 16:39 edm_services.client1_template
-rw-rw-rw-  1 root    other      812 Dec 20 16:20 edm_services.default
-rw-rw-rw-  1 root    other      816 Jan  9 16:40 edm_services.localhost
-rw-r--r--  1 root    other      812 Jan 10 09:42 edm_services.client2
lrwxrwxrwx 1 root    other      22 Dec 20 16:34 edm_services.edm1 -> edm__services.localhost
```

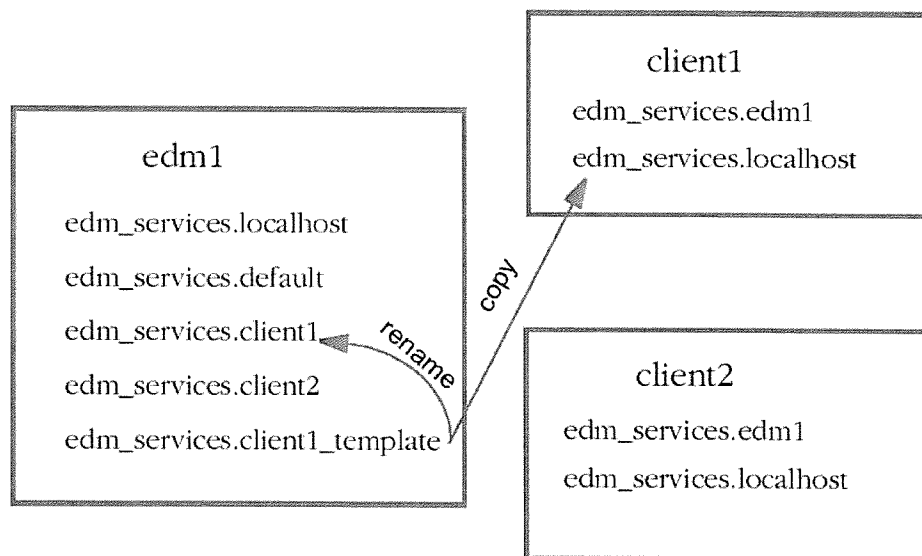
Note the file extensions in the examples.

1. The .default file is cloned to create the .client1 and the .client2 files.
2. These are then pushed out to client1 and client 2 to enable port control. They are stored on client1 and client2 as the .localhost file and on edm1 as the .client1 and client2 files.
3. The .localhost file on edm1 is pushed to client1 and client2 and named the .edm1 file.



To change the configuration of client1:

1. Use the **portservices** command (see the `portservices man` page) to create an `edm_services.client1_template` file with uncommitted changes on the server.
2. Reinstall client1, selecting yes in the port control window of the Backup Client Install Wizard.
 - a. This copies the `.client1_template` file to client1 and overwrites the `.localhost` file.
 - b. If that is successful, it will then rename the `.client1_template` file to `.client1` on the server and overwrite the existing file.



Note: If there is a `.client_template` file, it contains uncommitted changes.

.localhost File

On both the server and the client, the .localhost file contains settings for the local system.

To view these settings, on either the server or the client, enter:

```
# portservices -disp localhost
```

The following output appears:

```
Port Configuration from file <edm_services.localhost>
```

```
-----
Low Port Number: 8000
```

```
High Port Number: 8250
```

```
Transient Offset: 31
```

```
Fixed Ports      : 8000:8030 (31)
```

```
Reusable Ports   : 8031:8250 (220)
```

```
Socket Option(s): SO_REUSEADDR|SO_LINGER_10
```

```
Lookup Method    : EDM port services file
```

Service	Type	Ident	Offset	Comment
-----	-----	-----	-----	-----
emrpcd	RPC	390000	0	staging daemon (HSM)
vmdaemon	RPC	390001	1	vmdaemon
emsd_1	RPC	390003	3	migration daemon (HSM)
emsd_2	RPC	390004	4	migration daemon (HSM)
ebfsd	RPC	390007	7	ebfs daemon
07dbapicl	RPC	390008	8	EB Database API Daemon
hsmd	RPC	390009	9	HSM API Daemon
epcommd	RPC	390010	10	File Browser Daemon
edmlinkd	RPC	390011	11	EDM-Link Daemon
bamd	RPC	390012	12	Backup Activity Monitor
edmdispd	RPC	390015	15	Dispatch Daemon (EDM Restore)
07dbapi	RPC	390018	18	EB Database API Daemon for Sybase
tfsmd	Socket	390020	20	TFSM Calypso Daemon

Installing Port Control on the EDM Client(s)

After port control has been enabled on the EDM, select Install Client from the Backup menu of the EDM GUI. The Backup Client Install Wizard appears. Select the client(s) you want to install.

NOS Client Access Window

For Windows NT filesystem client(s), enter the username and password in the NOS Client Access window.

UNIX Client Install Method Window

For UNIX clients, proceed to the UNIX Client Install Method window. Follow the decisions you made in the Port Control Checklist.

Be sure that the local firewall administrator has implemented the firewall rules decided on in the Port Control Checklist. These ports may then be closed after the installation/update of the EDM client(s).

Note: The EDM Transfer Protocol, which requires a password, uses the UNIX REXEC protocol during UNIX client installation. The remote shell uses UNIX RSH during UNIX client installations. Both methods of installation use EDM Transfer Protocol during normal operations.

Note: If the port control configuration for the EDM .localhost file changes, all port-controlled clients must be updated to get the changes. (See "Making Changes to Port Control" on page 4-20.)

Port Control Window

When you reach the Port Control window, click:

- **Yes** to enable port control if it is not already enabled on the client. If port control is already enabled on the client this will overwrite the settings with the current settings on the EDM server.

- **No** to leave the port control settings on the client as they are. This will not enable, disable, nor update port control on the client.

Note: If you do not reach the Port Control window, the server does not have port control enabled. Enable the server using **eb_server_config** before proceeding.

Once a client has port control enabled and the configuration saved, it can be reinstalled without specifying port control, unless port control is removed from the client using the **portservices** command (see the portservices man page).

Note: You cannot install a client using an IP address, then move it behind the firewall, and give it a new IP address.

Making Changes to Port Control

If your port control configuration changes on the server, EDM clients should be reinstalled to specify the new settings. If changes have been made, the `edm_services.client_template` file appears on the EDM.

To Change System Monitoring for Ping Errors

Customers accessing clients through a firewall which does not allow ICMP packets to pass through in both directions may get RASD errors. While the default configuration for RASD is to check the availability of all clients, this setting can be modified through the System Monitor window in the GUI.

To Change the Default Port Range

If you want to change the default range, begin with the EDM, then reinstall the client(s).

1. To change the range from the default of 8000:8250 to 9000:9250, shut down the server, remove the old port configuration, make the changes, activate the changes on the server, and restart as shown below:

```
edm# edmproc -shutdown
edm# portservices -portconf default -low 9000 -high 9250
edm# portservices -activate
edm# edmproc -restart
```

2. Install the `edm_services` files on every port-controlled EDM client(s) as follows:

```
edm# portservices -portconf <client> -low 9000 -high 9250
edm# portservices -copyto <client>
```

An Alternative Method

Open the firewall for REXEC or RSH and reinstall the EDM client(s), selecting "Yes" in the Port Control window. This will change the ranges on the clients to match the new range on the EDM.

Close the firewall for REXEC or RSH.

To Enable Port Control with eb_server_config:

1. To determine if an EDM has port control enabled, enter **portservices -disp localhost**. If there are no edm_services files, the EDM does not have port control enabled.

2. If you want to enable it, do the following:

```
edm# edmproc -shutdown
edm# load_portfile [ options]
edm# portservices -activate
edm# edmproc -startup
edm# rpcinfo -p | grep 3900
```

3. **rpcinfo** displays the following:

390007	1	tcp	39552
390011	4	tcp	8011
390010	2	tcp	8010
390008	1	tcp	8008
390012	2	tcp	8012
390015	1	tcp	8015

4. Compare the right column for 390011, 390010, 390008, 390012, and 390015 with the settings shown by **portservices -disp localhost** (see “.localhost File” on page 4-17). For instance, 390011 has a low port number of 8000 and an offset of 11. Add these to get 8011 and compare to the output of **rpcinfo**.

To Turn Off Portmapper on the EDM

```
edm# edmproc -shutdown
```

```
edm# portservices -portconf localhost -lookup services
```

```
edm# portservices -activate
```

```
edm# edmproc -restart
```

To turn off portmapper on the EDM:

1. On the edm:

2. Install the changed edm_services files on all of the port controlled EDM client(s).

```
edm# portservices -copyto <every port controlled client>
```

An Alternative Method

Open the firewall for REXEC or RSH and reinstall the EDM client(s), selecting “Yes” in the Port Control window. This will change the ranges on the clients to match the new range on the EDM.

Close the firewall for REXEC or RCMD.

Note: To turn portmapper back on, use **-lookup portmapper** instead of **-lookup services**

To Turn Off Portmapper on Client(s)

To turn off portmapper on client(s):

On the edm, for each affected client:

```
edm# portservices -portconf <client> -lookup services
```

```
edm# portservices -copyto <client>
```

Note: To turn portmapper back on, use **-lookup portmapper** instead of **-lookup services**

To Set Portmapper Off for New Clients

To set the default for any new port control client, enter the following on the edm:

```
edm# portservices -portconf default -lookup services
```

Note: To turn portmapper back on, use **-lookup portmapper** instead of **-lookup services**

To Disable Port Control for a Single Client

To disable port control on a single client, do the following on the EDM:

```
edm# portservices -removefrom <clientname>
```

This removes the edm_services files from the client and saves the edm_services file for the client in a .template file on the EDM. If port control is enabled for the client at a later date, the .template file is used, thereby restoring the port control settings to what they were before port control was disabled.

To Disable Port Control on the EDM and All of Its Clients

If you want to disable port control on the EDM and all of its clients, do the following on the EDM:

```
edm# edmproc - shutdown
```

```
edm# portservices -removeall
```

```
edm# edmproc -startup
```

If any client(s) cannot be reached, the edm_services files on the client are not removed and must be removed manually. The edm_services files are located in the sub-directory etc/csc under the directory in which the EDM client software was installed (usually /usr/epoch).

5 How Backup and Restore Work

This chapter provides an in-depth description of what actually happens during the backup and restore processes. The following topics are discussed:

- How Backup Works
- How Restore Works
- Media Management

For information on how EDM Symmetrix Path backup and restore works, refer to the *EMC Data Manager Symmetrix Path User Guide*.

For information on how to perform EDM Symmetrix Connect backups and restores — specifically with the EDM Oracle Application Interface and Filesystem Application (for UNIX clients) — refer to the *EMC Data Manager Symmetrix Connect User Guide*.

For information on how to perform EDM Symmetrix Connect backups and restores with RMAN Proxy Copy (on UNIX clients), refer to the *EMC Data Manager Oracle Backup Client* guide. Similarly, see the *EMC Data Manager EMC Backint* guide for information on using Symmetrix Connect with the SAP R/3 System's SAP Tools (on both UNIX and Windows NT clients).

For information on how to perform EDM Symmetrix Connect backups and restores with the EDM-specific interfaces to NT filesystems and databases:

- *EMC Data Manager Windows NT Backup Client*
- *EMC Data Manager Windows NT Oracle Backup Client*
- *EMC Data Manager Windows NT SQL Server Backup Client*
- *EMC Data Manager Windows NT Exchange Backup Client*

How Backup Works

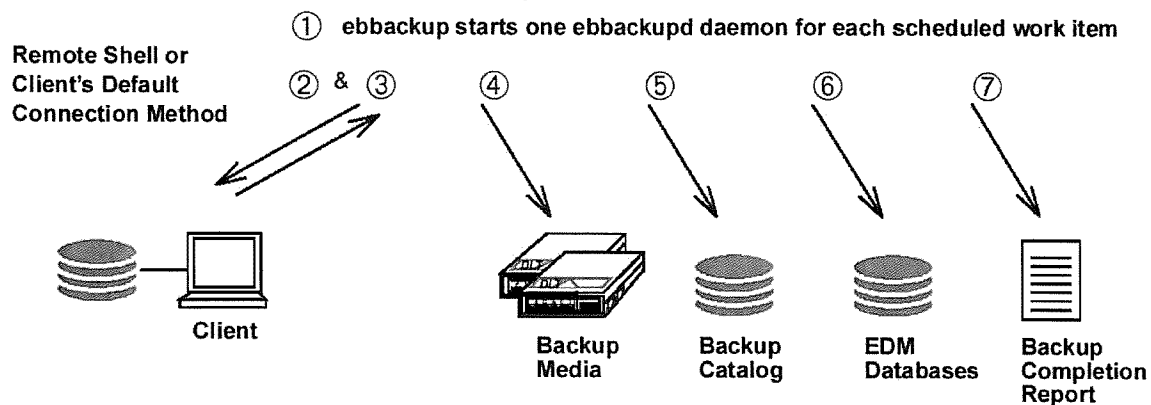
Use the EDM Backup Configuration window or command line interface to configure your filesystem and online database backups. The configuration process involves identifying and scheduling each client's data for backup. Once configured, backups occur automatically, copying the files from the clients' magnetic disks to the server's storage devices.

When client software is installed on the server, the magnetic disk(s) on the server are handled the same way as the local disks on client file servers and workstations. When the server is backing up its own local data, it is referred to as the local backup client.

The following is an overview of the backup process:

Figure 5-1

The Backup Process



1. The **ebbackup** program starts one backup daemon (**ebbackupd**) for each scheduled resource (work item).

The **ebbackup** utility is started from root's crontab file at a specified time.

2. The backup daemon connects with network clients via the client's connection method.

The backup daemon then tells the client what data to back up on that client and at what level (for example, full or incremental).

3. Remote and local clients call the **startfind** utility which starts a **findxcpio** process and then does a file scan.

The client scans its filesystems and sends the information (file attributes and the files to be backed up) back to the server.

4. The backup daemon on the EDM Backup server receives the information from the clients, stores it on backup media, creates a saveset record for the backup, and puts file name and attribute information in the backup catalog database.

5. As each work item is backed up, the catalog daemon, **ebcatalogd**, processes the backup catalog files for future use by the restore program.

6. The **ebbackupd** daemon updates several server databases, such as the volume management and saveset databases.

For HSM systems, **ebbackupd** updates the database so that it can determine the backup status of each client. The daemon updates the mtag.list to reflect correctly the mapping between work item names and EDM Backup/HSM tags; and, for baseline backups, it updates the saveset-to-baseline relations database to record the volume(s) and saveset ID of the backup. (For an understanding of HSM, read Part II "Hierarchical Storage Management.")

7. EDM Backup creates backup completion reports to inform you of successful and failed backups.

Monitoring Active Backups

You can monitor active backup processes and execute reports or queries in the EDM GUI. During a backup, an object in the Main window, such as the EDM server, a client, or a work item, appears as an active process, successfully backed up, in the backup queue, or failed to complete successfully. Current backup throughput also appears for an backup in progress.

Upon completion of a backup, you can then configure and save backup queries to report on specific areas of importance such as failed work items or work items with poor performance.

For more information about active backup monitoring and reporting, refer to EDM online help, “Backup Report Overview.”

Start of Overall Backup

The **ebbbackup** command, which is run out of root's crontab file, starts one **ebbbackupd** daemon for every scheduled work item. The **ebbbackupd** daemon performs the backup of a work item listed in the work group list of the backup template.

Backups of work items for multiple clients on the network can proceed concurrently. You can identify work items that back up the same physical disks so that they do not run at the same time. This prevents disk thrashing, thus improving the time to complete the backup. You can also assign a priority to each work item to control which ones are to be backed up first.

Client Access

The **ebbbackupd** daemon connects with network clients by using the client's connection method. Remote UNIX and PC clients use the EDM transfer protocol or the remote shell (**rsh**) utility; online database clients use RPC. The daemon accesses the local client (the server) directly. On UNIX clients, it invokes the **startfind** script on the client system.

Work Item Specification

The daemon provides the client with a specification of the filesystems, directories, and/or files to be backed up, as defined in the work item configuration. When doing this, the **ebbbackupd** command takes into account the priority at which each work item is processed; the level of completeness of the backup of an HSM system; and any exclusion tags (do not process work item A at the same time as work item B, etc.). For incremental backups, the daemon also takes into account the date/time of the last backup.

Automatic Scheduling

If a client that is listed in the backup schedule template's work group is unavailable, EDM Backup's autoscheduling function continues to back up all of the other clients that are listed in the work group. On the next day that the client is available, EDM Backup performs the backup.

Backup Activity Wizard

The Backup Activity Wizard enables you to start new, queued or failed backups, stop running backups, or manage the backup queue. You access this wizard from the Main window of the EDM GUI.

Note: You must have root privileges of be an EDM Backup Administrator to use the Backup Activity Wizard.

For more information about this wizard, refer to EDM online help.

Client/Server Processing Methods

EDM Backup consists of software that backs up the server and various network clients.

- EDM Backup's *server software* manages all networked client backups.

It provides central configuration and administration of your client backups. The server software also enables you or your users to restore files from backup media easily.

- EDM Backup's *client software* runs both on the server system (as a local client) and on each networked client.

When prompted by the server, the client scans its filesystems and sends the server the files to be backed up.

Refer to the *EMC Data Manager Software Release Notes* for a current list of clients.

Standard Client Processing

The client passes to the server the name and attributes for all scanned files, along with the data for those files that are selected for backup. The client sends the header information even for files that are not being backed up, to be able to reproduce the state of the filesystem at the time of the backup.

With the standard backup processing model, the client sends this information using the "standard output" channel of the connection, using an extended **cpio** format (provided as part of the EDM Backup software).

High-Speed Client Processing

For high-speed client processing, multiple data streams are generated. The header information, which contains the attributes, is sent in one stream to the server, which in turn writes the header data to a backup catalog. Another stream that contains the file data is sent to the backup media.

Server Processing

The server writes the header data to a backup catalog and sends the file data to the second server process, which in turn writes the data to the backup media.

Filesystem Backup

Filesystem backups are performed while you are online. Files are monitored as the backups are processed, so if a file changes while it is in the process of being backed up, the backup of that file is rejected and another is scheduled.

To reduce the backup workload, filesystem backups include *incremental* backup as well as full backups. A level-9 incremental backup backs up only those files that changed since their last backup. Each night, by default, the server software schedules full backups for some hosts and incrementals for the remainder. The scheduler rotates the full backups among all of the hosts over a rotation period, which by default is two weeks.

You can restore individual files from filesystem backups.

Baseline backups, which are available with the HSM option, back up your most stable files. From that point on, you perform backups relative to the baseline.

Refer to the *EMC Data Manager Symmetrix Connect User Guide* for information on backing up UNIX, Oracle, and Windows NT filesystems using EDM Symmetrix Connect.

Client Scans Filesystem

startfind and **findxcpio** are utilities that are part of the EDM Backup software. The **startfind** client script runs a filesystem scan utility called **findxcpio** to collect backup data from the client. This utility operates through the filesystem interface so it can work while the client's filesystems are active.

EDM Backup can detect that a file is being changed while it is being backed up. If **findxcpio** detects that a file changed, it backs up the file again. The **findxcpio** utility tries to copy the file up to three times before it skips it. Under these conditions, the file is backed up on the next scheduled backup.

For more information refer to Appendix D “findxcpio Directives”.

Database Backup

There are several ways to back up databases, as described in Chapter 6, “Database Backup and Restore” of this manual.

Note: For Symmetrix Connect, see also the *EMC Data Manager Symmetrix Connect User Guide*, the *EMC Data Manager Oracle Backup Client* guide, the *EMC Data Manager EMC Backint* guide, the *EMC Data Manager Windows NT Oracle Backup Client* guide, the *EMC Data Manager Windows NT SQL Server Backup Client* guide, and the *EMC Data Manager Windows NT Exchange Backup Client* guide.

ACL Support

An Access Control List (ACL) provides an enhanced level of security for UNIX files. ACLs extend the standard UNIX permission settings beyond owner, group, and other. An owner of a file can permit or deny access to specific users and groups.

For a list of platforms that support ACLs, refer to the *EMC Data Manager Software Release Notes*. HP, IBM, DEC, and Sun platforms implement ACLs differently. Refer to the appropriate client documentation for details.

Backing Up Files with ACLs

The backup software retains ACL settings when a file is backed up. During the backup process, backup writes the ACL to the media along with the data. When restored, backup properly restores the data with the same permission settings to the originating or same type client.

IBM clients support file ACLs up to one memory page (approximately 4096 bytes) in size. However, backup does not retain a file's ACL if it exceeds 1024 bytes.

If you attempt to back up a file that has an ACL larger than 1024 bytes, the backup process backs up the file without the ACL data. Only the standard UNIX file permissions are preserved.

This also produces an error message. If this error occurs often, consider adding more user groups to manage file access logically.

Restoring Files with ACLs

When you browse backup catalogs and mark files for restore, ACL settings are not visible in the file listing. However, backup checks the ACL settings and prohibits users who do not have permission to restore the file. A user can have access according to standard UNIX permissions but is prohibited from accessing a file if specified by the ACL. On the other hand, if a user has access to a file via the ACL but does not have standard UNIX permission, the user cannot mark the file for restore.

Due to the way that ACLs were implemented in Solaris 2.5.x, restore of ACLs on *directories* is not supported. (Restore of ACLs on files is supported.) The way Solaris 2.5.x implemented ACLs, the root account cannot change the ACL of a file or a directory of which root is not the owner. The restore software reconstructs directories from their attributes, but since no owner is defined, the restore software is unable to set the ACL.

Cross-Client Restore

The backup software does not support cross-client restore of ACLs. An ACL setting is retained only if you restore the file to the same platform type. For example, if you back up a file with ACL settings from an HP platform you can only restore the file with its original ACL to an HP platform. If you restore the file to another non-HP platform, the file is restored but the ACL is not retained.

Client ACL Commands

The commands that you use to list and set ACLs differ for each platform. Table 5-1 lists ACL user commands for the HP-UX, IBM AIX, Sun Solaris, and DEC UNIX platforms. Refer to the HP, IBM, Solaris, or DEC UNIX documentation for more information.

Table 5-1

Client ACL Commands

Operating System	Command	Description
HP-UX	chacl (1)	Change ACLs of files
	getaccess (1)	List access rights to files
	lsacl (1)	List ACLs of files
IBM AIX	acledit (1)	Edit an ACL
	aclget (1)	List ACLs of files
	aclput (1)	Set an ACL for a file
Sun Solaris	acl (2) *	Edit an ACL
	aclsort (3) *	Sort an ACL
	getfacl (1)	List ACLs for a file or files
	setfacl (1)	Set an ACL for a file or files
DEC Unix	getacl (1)	List ACLs of files
	setacl (1)	Set an ACL for a file or files

* System call or library function

Client Pacing

You can use Client Pacing to control the network bandwidth that the network backup clients of the EDM use. Enabling Pacing frees up computer resources for use by other

applications. Pacing then ensures that the average network utilization over a period of time does not exceed a specified threshold, thus “pacing” resources among applications.

Note: Client Pacing is available on all UNIX clients except Auspex and SunOS.

To use of the Client Pacing feature, do the following:

1. Make sure you install (or reinstall) the client after installing the EDM server software.
2. Create a file “pacer.cfg” on the client, in the directory /usr/epoch/EB. This file should have read permission for all users. This is a single line text file with the format:

Threshold [*debug_mode*]

where:

Threshold specifies the threshold value in KB/sec. The smallest permissible value for threshold is 100 (KB/s).

debug_mode is optional. Valid values are:

- a. “Verbose,” which writes Pacer trace messages to the file pacer.log in /tmp. Verbose is for temporary use; continued use could flood /tmp.
 - b. “quiet,” which suppresses Pacer trace messages (the default if no value is specified). For example, to limit the network utilization to 1000 KB/sec, this file should have an entry of 1000.
3. Run your backups as usual.

When the backup process starts on the client, it reads this file. If the file is successfully read and parsed, the pacing feature is enabled, and a message is sent to be logged in the file backups.log, in directory /usr/epoch/EB/log, on the EDM server:

Client Pacing is enabled for this backup. Threshold = 1000

where the threshold value is the one set in `pacer.cfg` on the client. You may easily disable the pacing feature by commenting out the `pacer.cfg` entry using “#” or removing the file `pacer.cfg` from the client.

Note: Understand that Client Pacing is done at the expense of the backup throughput. Overall backup performance is, by definition, impacted.

Keep in mind the following important points:

- The threshold value is applied to each backup that may be running for the client. It is NOT a collective threshold for all backups. So, for example, if threshold is set to 1000 KB/sec, and two backups are running concurrently for the same client, each is paced to the order of 1000 KB/sec, and the overall network utilization by all of the backup processes is 2000 KB/sec.
- Once the backup process begins on the client, the `pacer.cfg` file is not read again. Thus, any changes to this file do not affect any backups that are already running.
- The threshold value should be perceived as an approximation. The EDM client attempts to keep the average throughput over a period of time under the threshold value, but the network utilization at any particular instant is not guaranteed to be equal to the threshold.

Server Processes Attributes and Data

When the backup server receives the files from the client, the **ebbbackupd** daemon creates a backup saveset on the server to hold the contents of the backup data stream that it receives from the client system.

The **ebbbackupd** daemons can interleave savesets from different clients on the same piece of backup media, allowing many backups to occur simultaneously.

The allocation and use of backup media is managed by Volume Management through the *trail* concept. A trail is a collection of backup media of the same type, which you expand by allocating new media as needed. You specify the type of media to use for each backup level when you define the trail.

The **ebbackup** program can optionally alternate trails every other day. This enables you to segregate data between separate and identifiable sets of media, which makes it possible to store backups off site.

To restore files, the restore program (called from the Restore window in the EDM interface or from the **ebrestore** command) uses the backup catalogs, which are essentially a snapshot of a client's file names at the time of the backup. The EDM Restore window enables you to browse through the file names at any point in time, and to select individual files or entire filesystems to restore. For more information, refer to the section "How Restore Works" on page 5-16.

Catalog Processing

When a backup is first completed, the raw data for the associated catalog exists on the server, but the catalog daemon (**ebcatalogd**) must process the raw catalog before the restore program can use it. You can have catalog processing performed concurrently with backups, or you can schedule catalog processing for a later time so that this task does not slow down backups.

By default, **ebcatalogd** starts when the system boots. To start and stop processing, add the following commands to the crontab file:

```
/usr/epoch/EB/config/daemon_startup -ebcatalogd  
/usr/epoch/EB/config/daemon_startup -stop
```

Refer to the **ebcatalogd** man page for more information.

Server Database Update

The **ebbbackupd** daemon is responsible for maintaining several server databases, such as the saveset, volume management, and the catalog databases.

Report and Log File Generation

When a backup completes, you can configure and save backup queries to report on through the EDM graphical user interface (GUI), as described earlier. Through this reporting you can gather information about a backup such as its status, total throughput, total size of the backup, and total files that were backed up. (Refer to EDM online help, “Backup Report Overview” for more information.)

At backup completion, the system also generates either a backup completion report to confirm backup operations or a backup failure report. For examples, refer to “Backup Completion Reports” on page 16-29 and “Backup Failure Reports” on page 16-31 for more information.

The **ebbbackup** utility can email success and/or failure reports to the system administrator or to a configured list of users.

The **ebbbackupd** daemon records progress reports in the server log file of each work item as it is backed up.

Backup Completion Reports

Backup completion reports describe successful backups. These reports list the backup template name, the backup start date and time, the name of the backup trail, each client and what was backed up from it, the amount of kilobytes of client data in the backup, any clients that were unavailable for backup, the total number of clients, files, directories, and the backup completion date and time.

The server **eb_server_config** installation procedure creates the **mailok** script to which it passes the backup completion information. The script mails the reports to individuals who are responsible for backup operations, and/or writes them to a log.

Backup Failure Reports

Backup failure reports list the backups that require manual intervention to proceed. These reports contain only serious error messages, such as notifications of an interrupted backup or of a backup that could not start. Backup failure reports do not include errors from which EDM Backup automatically recovered.

The server installation procedure creates the **mailerr** script to which it passes the backup failure information. The script can mail the reports to individuals responsible for backup operations, and/or write them to a log.

Backup and Restore Logs

The **ebbackupd** daemon records progress reports in the server log file of each work item as it is backed up. During setup, a *logging level* is specified in each backup schedule template, which controls how much information this file contains. You can modify the default logging level, which is *stats*.

The `/usr/epoch/EB/log` directory on the server contains the following log files:

- `backups.log` — contains an audit trail of backup-related activities listed in chronological order. EDM Backup adds information to this file each time it backs up a template's work items. Selected notifications that appear in this log file also appear in other backup reports. As each log file accumulates information, EDM Backup removes the oldest ten percent of the data after the file reaches a configured maximum size.
- `recoveries.log` — contains an audit trail of restore-related activities that are listed in chronological order. EDM Backup adds information to this file each time it performs a file or database restore for a client.
- A log file for each backup template — contains the backup history for a single backup template. The information in this log varies depending on the logging level you specify in the

configuration database. Thus, you can use the file to view a history of backup-related events for a single template. The log files are named *template_name.log*.

For more information, refer to “Log Files” on page 16-35.

How Restore Works

The restore program provides a means of retrieving data from backups, which ensures that lost or damaged data can be quickly replaced.

Note: For Symmetrix Connect, see also the *EMC Data Manager Symmetrix Connect User Guide*, the *EMC Data Manager Oracle Backup Client* guide, the *EMC Data Manager EMC Backint* guide, and the various Windows NT client guides.

Table 5-2 gives you an understanding of the valid restore paths for the three types of backup paths.

Table 5-2

Backup versus Restore Path

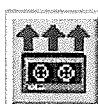
Backup Path	Restore Path		
	Network	Symmetrix Path	Symmetrix Connect
Network	Yes	Yes	No
Symmetrix Path	Yes	Yes	No
Symmetrix Connect	Yes ¹	Yes ¹	Yes

1. You cannot restore Symmetrix Connect data via network or Symmetrix Path if your Symmetrix Connect backup is a striped LVM configuration. You also cannot restore a Windows NT Symmetrix Connect backup via network or Symmetrix Path.

The restore program offers self-service file retrieval. You can configure the backup server so that users on UNIX client systems can perform file restore without the aid of a system administrator.

The EDM Backup server tracks user names and the clients from which these users have permission to restore files. This provides system security by enabling access control on a per client system basis. The software also enforces UNIX file permissions and Access Control Lists, if supported by the client. Users can restore only those files for which they have access permission.

Note: Refer to your *EMC Data Manager Software Release Notes* for a current list of client platforms that support ACLs.



Click this icon in the Main window of the EDM GUI to open the Restore window, or enter the command **edmrestore** at the CLI.

Use the CLI command **edmcrestore to open the Restore window on clients**. The EDM makes the connection and opens the Restore window. Alternatively, users can issue the command **ebcrecover** to restore files.

Note: The **ebcrecover** command on the client calls **ebrecover** on the EDM, but as of EDM 4.5.0, **ebrecover** is just a symbolic link to the actual restore program, **ebrestore**.

(Refer to the **edmrestore**, **edmcrestore**, and **ebcrecover** man pages for more information about these commands.)

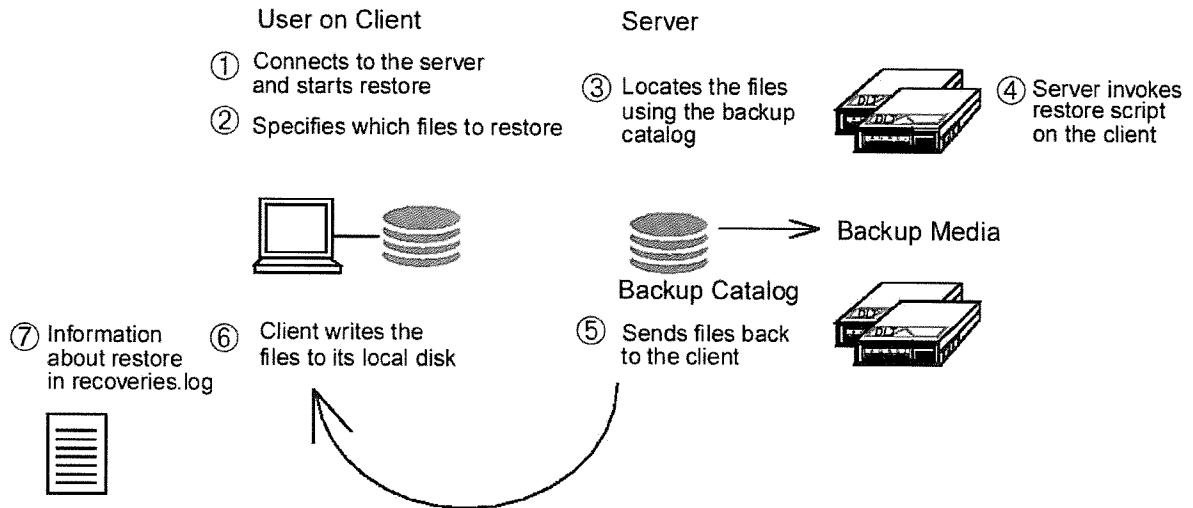
The restore program (**ebrestore**) runs on the server and enables you to extract data from backup media. Restore works with the help of on-line catalogs that keep track of all files that are saved on each backup volume.

Note: Try to avoid restoring a work item that is currently being backed up.

Figure 5-2 shows an overview of the file restore process:

Figure 5-2

The File Restore Process



1. A user on a client system issues the **edmcrestore** command to connect to the server and start up the restore window.

During one restore session, a user can restore files from different backup savesets. This causes the restore program to combine catalogs to reproduce the state of the filesystem at the time of the request.

2. A user searches for and marks files to restore. The server then locates the backup savesets that contain the specified files.

Note: When restoring offline database backups, you must select all stripes from the same backup date. You cannot restore one stripe, for example, from one backup date and another from a previous date.

3. The backup program scans the saveset records, which identify the backup catalogs, and locates the media that stores the backup data.

4. The **restore** program connects to the client using the client's native connection method (or direct connection in the case of a local client).

In either case, an EMC-provided restore shell script is invoked on the client system. The backup daemon provides the client script with a specification of the files to be restored.

5. The **restore** program reads the files being restored from the server's storage devices, packages the files in an extended **cpio** format, and sends them to the client system.
6. The client system writes the files to its disk.

If the files are being restored to the original client, the restored files overwrite any files that already exist on that client (unless the user specifies an option to not overwrite existing files). If the files are being restored to a different client, the server restore the files to the requested location.

7. Information about the restored files is written to the `recoveries.log` file on the client and on the server.

The `recoveries.log` includes the name of the user who is performing the restore operation, the total amount of data restored, and the date and time of the restore.

Media Management

The server's volume management software manages all media for backup and HSM (for network backups only). The software controls the library unit robotics that automatically insert media into and remove them from drives when an application requires their use.

For an understanding of media management, refer to Chapter 7 "Basic Volume Management Concepts" and Chapter 8 "How Volume Management Works."

Expiring Backups

As time passes, your site will have many volumes of backup data. If the site no longer needs to keep older backup data, it can specify that backups that are older than a certain age be automatically expired. When every backup with data on a volume has expired, the media is made available again. The volumes that are made available via expirations are made available only for their current trail. You may relabel these volumes to remove this restriction.

Deallocating Baseline Volumes

In HSM systems, baseline volumes do not expire directly but become deallocated (that is, available for reuse) when three conditions are met:

- The volume is no longer the current volume for the trail.
- The volume is not referenced by any baseline-relative backup.
- No block on the volume is in use; that is, no data is actively referenced by the current file system (as when a baseline ID in a file's metadata points to the volume).

6 Database Backup and Restore

Various options are available for database backup and restore. Backups can be either over the network or over SCSI or Fibre Channel cabling from a Symmetrix storage system. This chapter describes the several ways to back up databases.

Note: Prior versions of EMC Data Manager included an “offline” backup feature for Oracle, Sybase, and Informix databases — no option required. EDM no longer includes this feature. See “EDM’s Legacy “Offline” Database Backup Feature” on page 6-17.

This chapter includes the following topics:

- Varieties of Database Backup
- Various Database Backup Clients
- Database Network Backup Overview
- EDM Symmetrix Path Overview
- EDM Symmetrix Connect Overview
- EDM’s Legacy “Offline” Database Backup Feature

Varieties of Database Backup

EDM supports database backup and restore over networks or through attached Symmetrix storage units. Database support is provided by optional backup clients and by two Symmetrix options, EDM Symmetrix Path and EDM Symmetrix Connect.

The database backup clients provide network backup for the various major databases on numerous system platforms. In addition, all of the clients work with EDM Symmetrix Path (on the most popular system platforms). Finally, some of the clients work with EDM Symmetrix Connect as applicable, but EDM Symmetrix Connect also provides specialized backup interfaces in some cases to support database backup.

Whichever backup methodology used, the EDM centrally manages backup processing and media operation. Provided that adequate tape drives are available, EDM can run Symmetrix-related and network backups at the same time.

EDM's network backup is an effective solution for data centers with many small to medium-sized database and file servers. Network backups and restores are accomplished through ATM, FDDI, Fast Ethernet, or Token Ring network connection(s). The network backup function scales well, performs multiple backups at once, and centralizes backup and media management.

The EDM Symmetrix Path option offers channel-speed backup of large databases (and filesystems) over a data path through Symmetrix storage units to the EDM. Backups and restores are done over Fibre Channel or SCSI cables connected between the database host, the Symmetrix, and the EDM.

The EDM Symmetrix Connect option specifically addresses the backup needs of very large databases. Backups and restores are done over Fibre Channel or SCSI cables connected directly between the Symmetrix and the EDM, providing a direct backup of the database (or its mirrored-copy) on Symmetrix storage units to the EDM.

Database Backup Clients

EDM's database backup clients support various databases from major vendors, specifically: Oracle™, SAP R/3™ System (Oracle database), Sybase®, Informix™, Microsoft SQL™ Server, and Exchange. Unix and Windows NT platforms are supported. (See the *EMC Data Manager Software Release Notes* for a current list of supported client platforms and operating system versions.)

For the most part, these database backup clients provide backup and restore of database systems through interactions with the standard backup utilities provided by the database vendor for use with their databases. For example, the EDM Oracle Backup Client supports backups by Oracle7's Enterprise Backup Utility (EBU) and Oracle8's Recovery Manager (RMAN).

Depending on the database client, backup and restore operations are initiated from the database system, from the EDM, or from either system.

EDM Symmetrix Path

The EDM Symmetrix Path option is available for use with all the database backup clients. See the *EMC Data Manager Software Release Notes* for a current list of database versions and operating system versions supported for EDM Symmetrix Path.

Instead of streaming backup data over the network from the database host to the EDM, EDM Symmetrix Path streams the data over the Fibre Channel or SCSI cables between the database host, the Symmetrix, and the EDM. (The network is still used for control communication between the EDM and the client.) Once configured for EDM Symmetrix Path, a client's backups and restores can be switched between EDM Symmetrix Path and the network with a simple reconfiguration.

In supporting EDM Symmetrix Path, the database clients interact with the database backup utilities in the same way as they do for network backup. For example, the EDM Oracle Backup Client supports EBU and RMAN backups over Symmetrix Path.

EDM Symmetrix Connect

The EDM Symmetrix Connect option supports backup of a few key database systems, (Oracle, Microsoft SQL Server™, and Exchange) residing on Symmetrix storage units. It also supports backups of filesystems.

With EDM Symmetrix Connect backups, data on the Symmetrix (either the database or a mirrored-copy) is streamed over the cables directly from the Symmetrix to the EDM. The network is still used for control communication between the EDM and the client and for backup and restores of archived redo logs and control files.

EDM Symmetrix Connect backs up data for a few key database applications and filesystems. Two of its application interfaces are standard; the others are EDM-specific interfaces.

EDM Symmetrix Connect backs up UNIX Oracle databases through two standard interfaces:

- RMAN Proxy Copy (for some UNIX platforms), enabling backups and restores through Recovery Manager (RMAN)
- EMC Backint, SAP-certified interface (for some UNIX platforms), enabling backups and restores through SAPDBA

The RMAN Proxy Copy application supports Oracle's RMAN utility's Proxy Copy backups in conjunction with the EDM Oracle Backup Client. For RMAN Proxy Copy, backup operations can be initiated from the database system or the EDM; restores can be initiated from the database system only.

The EMC Backint application supports the SAP R/3 System's SAPDBA backups of Oracle databases, using the SAP R/3 System's standard BACKINT interface in conjunction with the EMC Backint client. For EMC Backint, backup and restore operations are initiated from the database system.

In addition, EDM Symmetrix Connect has two EDM-specific interfaces to Oracle:

- EDM Oracle Application Interface (for several UNIX platforms), enabling backups and restores through EDM
- EDM interfaces for Windows NT systems for Oracle, Microsoft SQL Server, and Exchange

With the EDM Oracle Application Interface, EDM Symmetrix Connect is tailored to work with directly with Oracle databases on the following client platforms: Compaq, HP, IBM, Sequent, and Sun. Rather than interacting with the Oracle backup utilities, the EDM Symmetrix Connect software provides its own interface to the Oracle database.

Similarly, EDM Symmetrix Connect's Windows NT interfaces support backups of Oracle, Microsoft SQL Server™, Microsoft Exchange™ (as well as filesystems on Windows NT). See the *EMC Data Manager Software Release Notes* for a current list of supported client platforms.

For the EDM-specific interfaces, backup and restore operations are initiated from the EDM.

Various Database Backup Clients

EDM supports various, distinct database clients that are packaged separately as options.

Oracle Backup Client

The EDM Oracle Backup Client, which interacts with the Oracle backup utilities, is available for many UNIX client platforms and Windows NT. It supports network, EDM Symmetrix Path, and EDM Symmetrix Connect (RMAN Proxy Copy) backup. This option is installed and configured through the EDM graphical user interface (GUI). See *EMC Data Manager Oracle Backup Client* for installation and configuration instructions for UNIX platforms. For Windows NT, see the *EMC Data Manager Windows NT Oracle Backup Client* manual. Also see the online Help in the GUI.

EMC Backint Client for SAP R/3 Oracle Databases

The EMC Backint client, which is a software interface to the SAP R/3 System for backing up its Oracle database, is available for many UNIX client platforms and for Windows NT. It supports network, EDM Symmetrix Path, and EDM Symmetrix Connect backup. This option is installed and configured through the EDM GUI. EMC Backint has its own client manual with specific installation and configuration instructions. See the *EMC Backint* manual.

Other UNIX Database Backup Clients

Sybase and Informix databases each have corresponding database clients, with individual client manuals with specific installation and configuration instructions, which are available for most client platforms. They support network backup and EDM Symmetrix Path. These options are installed and configured through the EDM GUI. See the *EMC Data Manager Sybase Backup Client* and *EMC Data Manager Informix Backup Client* manuals.

Microsoft Database Backup Clients

Microsoft SQL Server and Microsoft Exchange each have corresponding database clients, with individual client manuals with specific installation and configuration instructions. They support network, EDM Symmetrix Path, and EDM Symmetrix Connect backup. These options are installed and configured through the EDM GUI. See the *EMC Data Manager Windows NT SQL Server Backup Client* and *EMC Data Manager Windows NT Exchange Backup Client* manuals.

Database Network Backup Overview

Database backups can be initiated from within the database management system or on the EDM. In the first case, the database administrator starts the backups from within the DBMS's own backup utility. In the second case, the EDM's scheduling function starts a process that, in turn, automatically starts the backups within the DBMS's own backup/restore utility.

Table 6-1 lists the backup and restore utilities for each database system.

Table 6-1

Network Database Backup and Restore Utilities

Database System	Database's Backup/Restore Utility
Oracle8	Recovery Manager (RMAN)
Oracle7	Enterprise Backup Utility EBU (obackup)
Oracle under SAP R/3	SAPDBA (BRBACKUP, BRARCHIVE, BRESTORE)
Sybase	dump/load
Informix	On-BAR

Once started, the DBMS's backup utility scans the database for data to back up and passes the data to the backup client software. Then the backup client software streams the data to the EDM over the network.

Multiple Streams

To speed up backup processing, you can create multiple streams of backup data from the database (for example, six). You can also decide to write the backups to multiple tapes (for example, two). The difference in the two example numbers represents a consideration of the generally slower speeds of magnetic disks (which hold the database) as compared with tape drives (which write the backups to tape).

Note: The various implementations of streaming for each database client are described in their respective manuals (or release notes, as applicable).

Server-side Processing

The server software manages the writing of data to tape storage media and provides online catalogs, located on EDM's own disks. Separate paths for data flow and for control (catalog information), allows the data to take a more direct path to the backup media.

The catalogs enable restores of your data from the backup media. (In the case of the Backint SAP R/3 client, the client software also creates catalogs at a meaningful granularity for the database system and stores them locally on the client.)

The server software also manages the operation of robotic library units and provides overall volume life-cycle management. Also, the software enables automated backup expiration.

Restores

The client software receives restore requests from the database's restore utility at a database, tablespace, or data file granularity, as appropriate, and sends the requests to the server software.

The server software retrieves the data from the backup media and sends it to the client software, which passes it on to the database's own restore utility.

User Interfaces

The backup clients can be installed through the EDM Backup Install Wizard and configured through the EDM Backup Configuration Wizard. They can be reconfigured through either the EDM Backup Configuration Wizard or the EDM Backup Configuration window. Some command-line procedures might also be required.

See the appropriate EMC manual for information on installing clients, configuring backups, and restoring data.

Note: Although the Restore window might display work items for the following database products, it does not support restores for: Oracle Backup Client, Sybase Backup Client, Informix Backup Client, and EMC Backint. Restores are accomplished through each database's restore utility.

Configuring Backups to an Alternate Network

Multiple-networked backup, meaning that if the *client* machine has multiple interfaces, the client can send backup data through those multiple interfaces.

You can specify which interface to use by Work Group/Schedule, using the Use Client Name parameter in the Work Item tab's Work Item Options window (in the Generic tab). This parameter corresponds to the `connection via` parameter of the "listener" work items in the `eb.cfg` file.

Note: A separate issue is the use of multiple network interfaces on the *EDM*. For database backup clients only, the client must have a valid EB server hostname in its `/usr/epoch/EB_DB/ebci.conf` file.

Database Pre-Discovery

During initial client configuration from the Backup Configuration Wizard, the client machine is scanned for the presence of databases, in a process called *pre-discovery*.

Pre-discovery reveals just two pieces of information:

- Database type (that is, Oracle, Sybase, Informix)
- Database name

No special requirements exist for Oracle databases to be pre-discovered; the Backup Configuration Wizard presents the databases listed in the `/etc/oratab` file as the databases that exist on the system. However, for Sybase and Informix databases, there are assumptions as to where the database software was installed, as described in Table 6-2.

Table 6-2 Requirements for Database Pre-Discovery

Database System	Requirement for Pre-Discovery
Oracle	<p>None. The Backup Configuration Wizard presents the databases listed in the <code>/etc/oratab</code> file as the databases that exist on the system.</p> <p>Note: If the file lists a database that actually does not exist, it is shown anyway, as pre-discovery is not able to ascertain whether in fact the database does not actually exist or if it is just off line.</p>
Sybase SQL Server	<p>One of the following on the Sybase machine:</p> <ul style="list-style-type: none"> • A link, <code>"/SYBASE"</code>, to the directory in which the Sybase database software is installed. • The environment variable <code>"SYBASE"</code> for the root environment pointing to the directory in which the Sybase database software is installed. • A UNIX account named <code>"sybase"</code> in whose home directory (as specified in <code>/etc/passwd</code>) the Sybase database software is installed.
Informix	<p>Either of the following on the Informix machine:</p> <ul style="list-style-type: none"> • The environment variable <code>"INFORMIXDIR"</code> in the root environment pointing to the directory in which the Informix database software is installed. • A UNIX account named <code>"informix"</code> in whose home directory (as specified in <code>/etc/passwd</code>) the Informix database software is installed.

EDM Symmetrix Path Overview

The EDM Symmetrix Path feature currently works with the EDM Oracle Backup Client, the EMC Backint client, Sybase client, Microsoft NT SQL Server client, Microsoft Exchange client, and Informix client.

EDM Symmetrix Path enables an EDM to back up your database (and filesystems) through direct SCSI connections to a Symmetrix, rather than over a local area network.

With this methodology, the Symmetrix itself acts as the network. A few small devices are designated as transport paths for configuration purposes, while the actual data transport is generally handled by the cache on the Symmetrix corresponding to these devices.

Note: As they are dedicated to transport, these devices are unavailable for use as storage devices.

These Symmetrix Transport Groups (also referred to as ST groups or STGs) are mapped to the hosts (clients and the EDM). Various possible device configuration and host mapping combinations provide different performance characteristics and degrees of flexibility.

Note: The database can be located on the Symmetrix on volumes that are accessible to the database host. But unlike with the EDM Symmetrix Connect methodology, the database volumes are not made accessible to the EDM, nor are mirror-images of the database volumes.

Each of the various possible device configuration and host mapping combinations provide different performance characteristics and degrees of flexibility. See the *EMC Data Manager Symmetrix Path User Guide* for more information.

EDM Symmetrix Connect Overview

With EDM Symmetrix Connect, databases are located on one or more Symmetrix systems in various configurations that are visible to the EDM. The backup data is sent over SCSI or Fibre Channel cabling that directly connects the EDM to the Symmetrix storage. The backup is usually taken from a mirrored copy of the database on a Symmetrix (three different mirrored-volume configurations are available), but a non-mirrored configuration, which backs up the database itself, is also supported.

Applications

EDM Symmetrix Connect's EDM Oracle Application Interface supports backup of Oracle databases for UNIX (Sun, IBM, HP, Sequent, and/or Compaq clients). The Filesystem Application of EDM Symmetrix Connect adds the capability to back up selected filesystems on UNIX systems.

Other EDM Symmetrix Connect applications support Oracle backups through RMAN Proxy Copy and through EMC Backint (for SAP R/3 System Oracle databases). Both online and offline database backups can be performed.

EDM Symmetrix Connect's Windows NT applications support Oracle, Microsoft SQL Server, and Microsoft Exchange (as well as Windows NT filesystems).

User Interfaces

On the EDM, the EDM Backup Configuration Wizard is used for client configuration of EDM Symmetrix Connect.

The Backup Activity Wizard or EDM's command-line interface is used for EDM-initiated backup.

Restores of EDM Symmetrix Connect backups taken through the EDM Oracle Application Interface are EDM-initiated using the command-line. Restores of Oracle RMAN Proxy Copy backups

are always performed on the client, through RMAN. Restores of EMC Backint backups are always performed on the client, through SAPDBA or BRRESTORE.

The EDM GUI's Library Manager window is available for media management operations.

Documentation

See the *EMC Data Manager Symmetrix Connect User Guide* for detailed information on how to perform each operation for the EDM Oracle Application Interface and for the filesystem applications.

See the *EMC Data Manager Oracle Client* guide for RMAN Proxy Copy, the *EMC Data Manager Backint* guide for EMC Backint, and the *EMC Data Manager Windows NT Oracle Backup Client* guide for NT Oracle.

See the other Windows NT client guides corresponding with the EDM Symmetrix Connect backups of your Windows NT system.

Raw Device Backups

For the most part, EDM Symmetrix Connect performs its database backups at the physical disk, that is, raw device level. Raw device backups have the advantage of fast backup performance. Their limitation is that they do not offer the granularity of file-by-file (logical) backups and restores.

If your database files are built on raw devices, than the raw device backups are perfectly suited.

If your database files are in filesystems, logical filesystem backups are available if your database server (the backup client) is running the same operating system as the EDM.

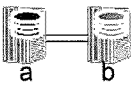



Otherwise, raw device backups of database files in filesystems will give you fast backup performance but also a lack of file-by-file granularity in backups and restores.

Configurations

Databases can be located on one or more Symmetrix systems in various configurations. EMC offers a direct connect backup and restore solution in a combination with four Symmetrix volume mirroring configurations as shown in Table 6-3.

Table 6-3

EDM Symmetrix Connect Mirroring Configurations

Backup Solution	Symmetrix ¹ Configuration	Description
Symmetrix SRDF Connection		Backup of “target” (“R2”) volumes on a second, connected Symmetrix (b), which mirror the “source” (“R1”) volumes on the first Symmetrix (a), which contain the database
TimeFinder		Backup of “Business Continuance Volumes” (“BCVs”) (b), which mirror the “standard” (“STD”) volumes (a), which contain the database, all of which are on one Symmetrix
Remote BCV		Backup of “BCVs” (c) on second Symmetrix, which mirror “target” (“R2”) volumes on the second Symmetrix (b) that mirror the “source” (“R1”) volumes on the first Symmetrix (a), which contain the database
Non-mirrored		Backup of non-mirrored volumes (a) containing the database

1. For all EDM Symmetrix Connect solutions, the Symmetrix models are 3xxx/5xxx ESP model systems.

Mirrored Configurations

The first three configurations listed in Table 6-3 take advantage of volume mirroring, which is when one (or more) exact copies of the database's disks are maintained simultaneously.

One mirroring capability is between two physical Symmetrix systems (the Symmetrix SRDF configuration). The other mirroring capability is between disks on a single Symmetrix (the TimeFinder configuration). The Remote BCV configuration uses both mirroring capabilities together.

To perform the backups, the EDM logically discontinues the active mirroring between the database disks (or the target "R2" volumes, in the case of Remote BCV) and the mirrored copies. The database host can continue to function normally using the primary Symmetrix disks while the EDM backs up the (now static) mirror-copy of the database (or tablespaces) rather than the database itself. At the time the mirrors were split, the two volume sets were exact copies of each other.

Backups can be online or offline. In a mirrored configuration, when performing online backups, the database is put into Oracle's online backup mode only briefly, just for the time the mirrors are split. (If any inconsistencies occur while the mirrors are split, they can be resolved by the Oracle software at restore time.) Therefore, for these mirrored configurations, the database host (client) is not impacted by the online backup operation.

When performing offline backups with the EDM Oracle Application Interface, the database or tablespace remains offline only briefly, just for the time the mirrors are split. However:

- For EMC Backint, the database or tablespace remains offline for the entire duration of the backup.
- For RMAN Proxy Copy, the database, tablespace, or datafile remains offline for the entire duration of the backup unless a post-mirror-split script is employed to bring the database, tablespace, or datafile online after mirrors have been split.

Non-Mirrored Configuration

In the Symmetrix non-mirrored configuration, the actual client's database is backed up, not a mirrored copy of it. To use this feature, the Oracle data files cannot reside in a filesystem. The data files must be in raw partitions.

The database can be backed up either online or offline. When performing online backups in a non-mirrored configuration, the database is put into Oracle's online backup mode for the entire duration of the backup.

Note: The extended time for online backup mode may be an issue for certain customers since Oracle overhead can be significant, if major updating is being performed during online backup mode. The effects of online backup mode overhead can be reduced by configuring backups, such that fewer tablespaces are requested per backup run.

When performing offline backups, the database or tablespace remains offline during the entire duration of the backup.

EDM's Legacy "Offline" Database Backup Feature

EMC Data Manager formerly included an "offline" database network backup feature for Oracle, Sybase, and Informix databases, for which no option was required.

Support for performing backups with this feature has been removed.

However, EDM does continue to support restores for any backups taken with this functionality using prior versions.

Also, EDM continues to support backups of these databases on most of the same client platforms through the use of EDM's Oracle, Sybase, or Informix backup clients.

7 Basic Volume Management Concepts

Volume management software manages and controls access to attached library units and all removable media that EDM Backup and the optional HSM software use.

This chapter describes the components that comprise the volume management software.

The topics in this chapter include:

- Volume Management Overview
- EDM Library Unit Manager
- Volume Manager
- Library Managers

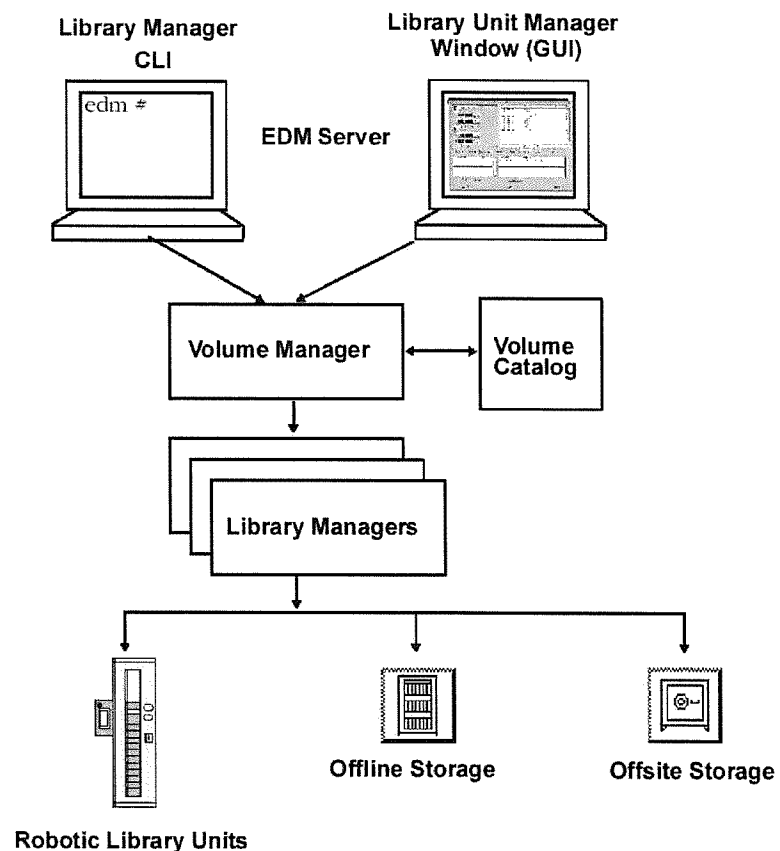
Volume Management Overview

The volume management software provides removable media management services to Backup and optional HSM software. The software includes the following components:

- EDM Library Manager (GUI and CLI)
- Volume Manager
- Device-specific Library Managers

Figure 7-1

Volume Management Software



EDM Library Unit Manager

The EDM Library Unit Manager, which resides on the EDM Backup server, is your interface to volume management functions. For example, you use the EDM Library Unit Manager to:

- monitor volume activity on the EDM server
- inject and eject volumes from the library unit
- label volumes for use
- locate available volumes
- check for outstanding media requests
- initiate partial or complete library unit inventories

The interface interacts with the underlying software components, which the following sections describe.

Volume Manager

The Volume Manager manages:

- information about all volumes
- volume life cycle
- volume requests that applications make (for example, backup and media duplication)

The Volume Manager maintains a group of files on the server in the directory `/usr/epoch/etc/vm`. This directory contains a configuration file (`vm.cfg`), the volume catalog (`volumes`), template catalog (`templates`), log file (`clog`), and other administrative files.

Note: With the exception of `vm.cfg`, volume management uses all of the files in this directory internally. You should not edit any of these files.

Volume Catalog

Volume management identifies all volumes by a unique electronic volume label on the media. The Volume Manager keeps track of all volume information in the volume catalog. The catalog contains entries for each volume including the volume ID, volume sequence number, physical location (by Library Manager), volume state, optional barcode ID, and usage count.

The Volume Manager updates the catalog as operations and events occur, such as when:

- new media enters a library unit
- a volume is allocated to an application
- volumes in a library unit are inventoried
- a volume is ejected from the library unit
- a volume is imported from another server
- a volume is moved from offline to offsite

The volume catalog is an integral part of volume management software and necessary for rebuilding an EDM system.

Therefore, the catalog as well as all volume management system files are backed up as part of the default server work group.

Volume Life Cycle

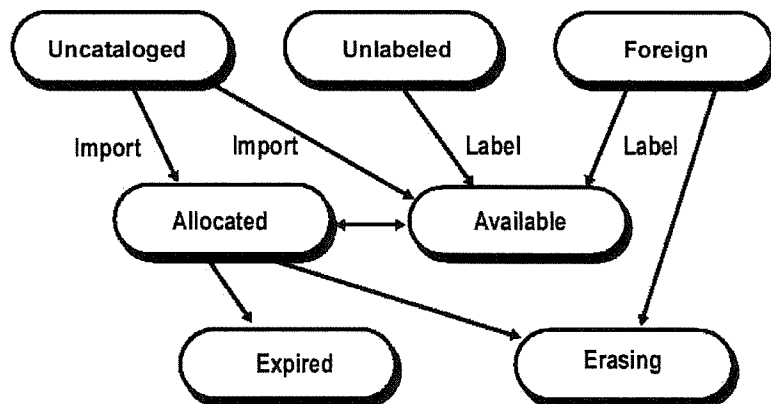
A volume's life cycle (as illustrated in Figure 6-2) begins when you label a new tape or optical disk. The labeling process writes a unique electronic label to the media. A labeled tape, such as DLT, holds one volume. Two-sided media, such as EO disks, contain two volumes, one volume for each media side.

When media is loaded into a library unit, its volume label is read before information for the volume becomes visible in the EDM Library Unit Manager. A volume is identified by its

sequence number or one of the following states: uncataloged, unlabeled, unverified, foreign, expired, or erasing (EO disks only). These states are described below.

Figure 7-2

Volume Life Cycle



Uncataloged

An *uncataloged* volume indicates that the volume has a valid volume label but is missing from the volume catalog. This can happen if the volume was labeled on another server or if the volume catalog was lost as a result of a server disk crash (see Chapter 20 “Recovering a Server from a Disk Failure”).

If you import the volume, the Volume Manager adds an entry to the receiving server’s volume catalog. The volume retains its volume ID and volume sequence number that the Volume Manager of the originating server assigns.

Volume management does not allow duplicate volume sequence numbers of the same media type. If you attempt to import a volume that has a sequence number that already exists, you are prompted to either delete the existing volume or cancel the import operation. (Refer to “Duplicate Volume

Sequence Numbers” on page 8-26 for instructions on how to override this restriction.) Volumes that are contained on different media types (for example, EO and DLT) may have the same sequence number.

Unlabeled

An *unlabeled* volume means the volume’s label area is blank. This is the state of new media. You can either label the volume or leave it unlabeled until an application makes a request for a new piece of media.

When you insert an unlabeled volume into a library unit, the Library Manager assigns it a slot number, moves the volume into the slot, and notifies the Volume Manager to add an entry for the unlabeled volume to the volume catalog.

Foreign

A *foreign* volume is any previously-used media from a non-EDM system. A **tar** tape is an example.

To reuse a foreign volume, you must first label it. When a foreign volume is labeled the Volume Manager adds it to the volume catalog and changes its state to available. If the media is an EO disk, the data on the disk is erased before it is labeled.

The EDM Library Unit Manager also allows you to mount foreign volumes for the purpose of reading or extracting data. You can do this by manually (or force) mounting the volume into a drive using the Force Mount button in the Utilities tab of the EDM Library Unit Manager window.

Note: You must dismount the volume manually to avoid preventing other processes from using the drive.

Expired

Tape media expires when it reaches a pre-set maximum usage. After backup deallocates a volume, the Volume Manager checks the usage count and expires the volume after the maximum is reached. (See Figure 7-3 on page 7-9.) When a volume expires, it cannot be mounted for data access.

Erasing

Erasable optical disks have an erasing state that occurs at different stages. EO disks enter the erasing state before a foreign EO is labeled and after HSM deallocates the volume. (See Figure 7-4 on page 7-10.)

Unverified Volume

An *unverified* volume means that the Library Manager is unable to recognize the label contents on the volume. An unverified volume can result for one of many reasons: the volume was just injected and did not yet complete the initial label read; the media is incompatible, an error occurred while the label was being read; a hardware problem occurred; or a user placed several volumes into a library unit (LU) through the mass load door and then ran a barcode-only inventory.

Generally you should inventory an unverified volume so that its label is read properly. For a library unit that supports barcodes, perform a barcode and label inventory; for a non-barcode LU, perform a label inventory.

Note: A barcode and label inventory is recommended.

If a drive becomes dirty while the volume is in the drive and its label is being read, the Library Manager automatically dismounts the volume, disables the drive and marks it as dirty, and tries to read the volume's label in another drive. If this second drive also becomes dirty, the Library Manager dismounts the volume, places it back in its slot, and marks the volume as unverified and offline. The Library Manager then disables this second drive and marks it as dirty, and places it back into the LU.

If the LU contains a cleaner cartridge, the drive is cleaned automatically; if no cleaner resides in the LU, injecting a cleaner starts an automatic cleaning. You must then verify the unverified volume either through a mount request or an inventory of that slot (barcode/label or label inventory).

Note: If the volume remains unverified after an inventory, remove it from the library unit.

How Volumes are Allocated

A volume becomes available for allocation after it is labeled. Labeling a volume requires that you choose a volume template. Volume templates enable you to specify whether a volume should be made available to any application (that is, Backup or HSM), to any trail, or to a specific trail.

Several volume templates are available:

- Unrestricted (*media_type*) — any application
- EBprelabel — backup only
- HSMprelabelEO — HSM only
- Restricted to *trail_name* — specified trail name only

The templates also contain attributes that the volume inherits such as: a unique volume ID, trail name, media type, maximum usage count, and media size information.

When a Volume is Allocated

Volume allocation begins with a request from backup, media duplication, or HSM. The Volume Manager locates an available volume based on the application's request, provides a volume ID, gives the application access to the volume, and changes the volume state to Allocated.

Volume usage is based on the number of times that a volume transitions from *Available* to *Allocated*. Each time the volume is allocated, the Volume Manager increments the media's use count by one. When a volume reaches its maximum usage, the volume is expired.

The figures on the following pages illustrate the volume life cycle of two media types. Figure 7-3 illustrates the life cycle of tape, Figure 7-4 on page 7-10 illustrates the life cycle of erasable optical media, and Figure 7-5 on page 7-11 illustrates the life cycle of WORM (Write One Read Many) optical media.

Figure 7-3

Tape States

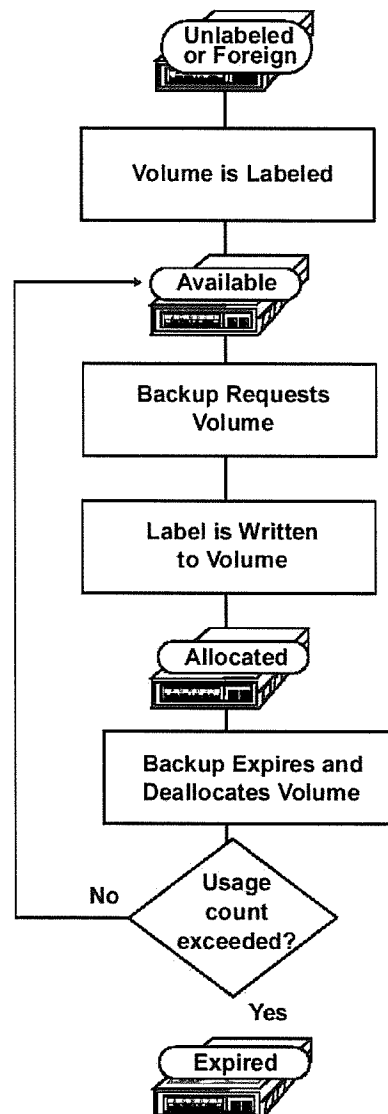


Figure 7-4

EO Volume States

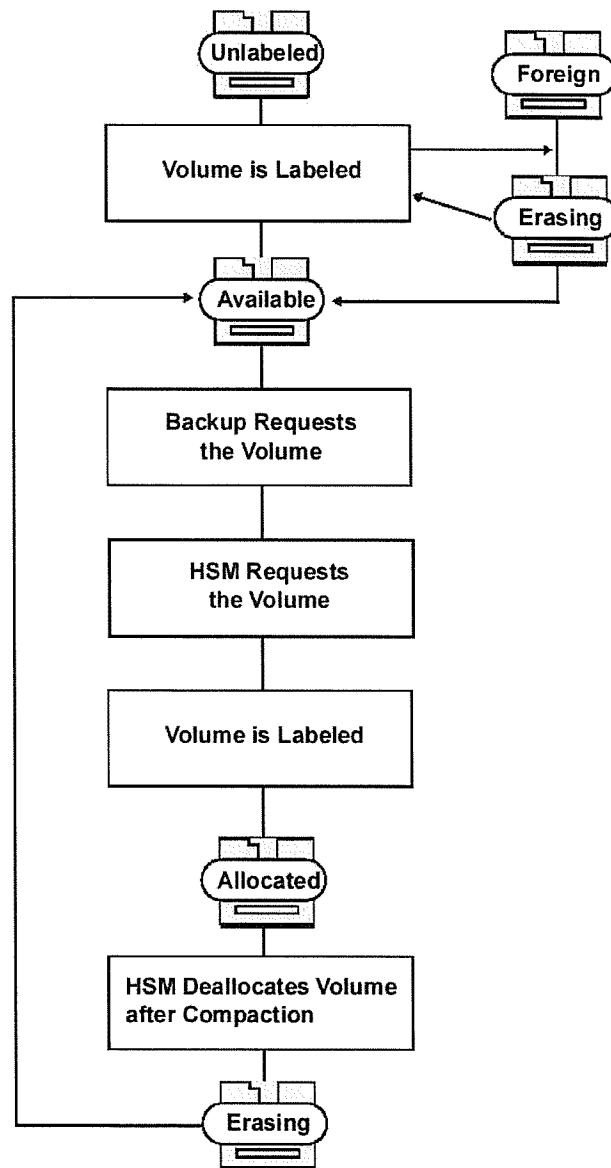
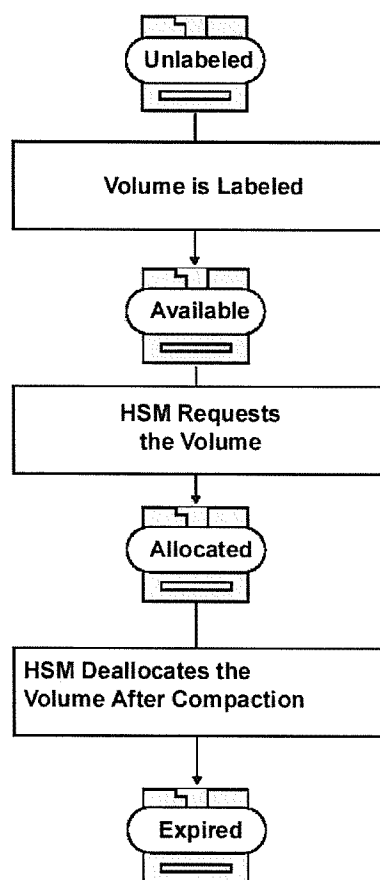


Figure 7-5

WORM Volume States

Library Managers

Device-specific Library Managers control library unit operations:

- library unit inventories
- drive preemption
- robot movement for mounting and dismounting media
- injecting or ejecting media

Offline and offsite Library Managers hold information about volumes in offline and offsite locations. Library Managers are supplied for various types of library units.

Library Manager Configuration

You use the **lmconfig** utility to configure a Library Manager for each library unit that is attached to the server. For each Library Manager that you configure, **lmconfig** sets up a subdirectory in `/usr/epoch/etc/lm` to include the configuration file and other internal files that the Library Manager uses. Each subdirectory name is based on the vendor name and model number.

Within the Library Manager's subdirectory, a configuration file (`lm.cfg`) defines features and functionality for the device. For example, if a library unit is equipped with a barcode scanner, the configuration file enables barcode support.

After a library unit is configured, an icon for the library unit and each internal drive appears in the Library Units and Drives area of the Library Unit Manager window (only if volume management is running).

Robotic Library Units

A Library Manager controls the library unit's robot (the mechanism that moves cartridges within a library unit), internal tape drives, and media inlet (if present).

Each Library Manager maintains a per-library unit inventory of volumes in the file `valid.dat` within its subdirectory. When a Library Manager is started for the very first time, it takes a complete inventory of the library unit's contents and creates the `valid.dat` file. Once the file is created, the Library Manager reads `valid.dat` to initialize the library unit, which eliminates a complete inventory each time the system is started.

The inventory list includes a volume ID, barcode label (if supported by the library unit), slot number, and drive location for each volume. As volumes move from one location to another, the Library Manager updates the inventory list and notifies the Volume Manager of any changes.

The Library Manager also controls drive scheduling and drive selection. When the Volume Manager makes a request for an operation (for example, a mount request) the Library Manager adds the request to a prioritized work queue. When a drive becomes available, the Library Manager services the next work item with the highest priority.

Offline and Offsite Library Managers

Offline and offsite Library Managers enable you to track the location of volumes that are outside of a physical library unit.

Offline represents volumes that are ejected from a library unit and stored in a nearby area, usually somewhere on site. The offsite Library Manager holds information about volumes that you physically move to a location beyond the building's boundaries, such as an offsite archival location. Only volumes that have a volume label or barcode label can enter the offline or offsite Library Manager. An unlabeled volume with no barcode is deleted from the volume catalog when it is ejected from the LU.

The EDM Library Unit Manager window displays an icon for offline_0 and offsite_0 Library Managers. From this window, you can view the volumes that are contained in both the offline and offsite Library Managers. You can also eject volumes into either the offline or offsite Library Manager using the Eject tab in the Library Unit Manager window. (Refer to EDM Online Help for instructions.) The Utilities tab in this window provides a text field that enables you to record the volume's actual offline or offsite location.

Note: Moving backup media offsite before its rotation period ends causes the backup that would use that media to fail. You can avoid a failed backup by using new media for the next backup. You can configure the use of new media in the Backup Configuration window of the EDM GUI. Select Advanced Options in the Schedule Tab. In the Schedule Options window that appears, select Use New Media When Current (backup media) Is Offsite.

The offline_0 and offsite_0 configuration files specify the action to be taken when a mount request is received for a volume that is offline or offsite.

Ejecting a Volume

When you eject a volume from a library unit by using the Eject tab in the Library Unit Manager window, the default destination is offline. (Volume Manager determines where to put the volume by the value of LM_EJECT_DEST that is set in the vm.cfg file.)

The Eject tab also enables you to eject volumes to offsite. Knowing whether a volume is located offline or offsite is helpful when you need to locate a volume for an application such as a restore. If the volume is not in any library unit, the Media Request window specifies whether the volume is offline or offsite.

8 How Volume Management Works

The information in this chapter is intended for the system administrator who wants in-depth knowledge of the volume management software. It describes the volume management processes and how they work together to provide services to EDM Backup and HSM software.

This chapter covers the following topics:

- rvmoper UNIX Group
- Volume Management Processes
- Volume Management Startup
- Library Unit Operations
- Volume Allocation and Deallocation

rvmoper UNIX Group

Normally, users who are not root can only monitor volume management activity. However, if you are a member of the rvmoper UNIX group (/etc/group), you can perform volume management functions, such as labeling media, and injecting and ejecting media from a library unit.

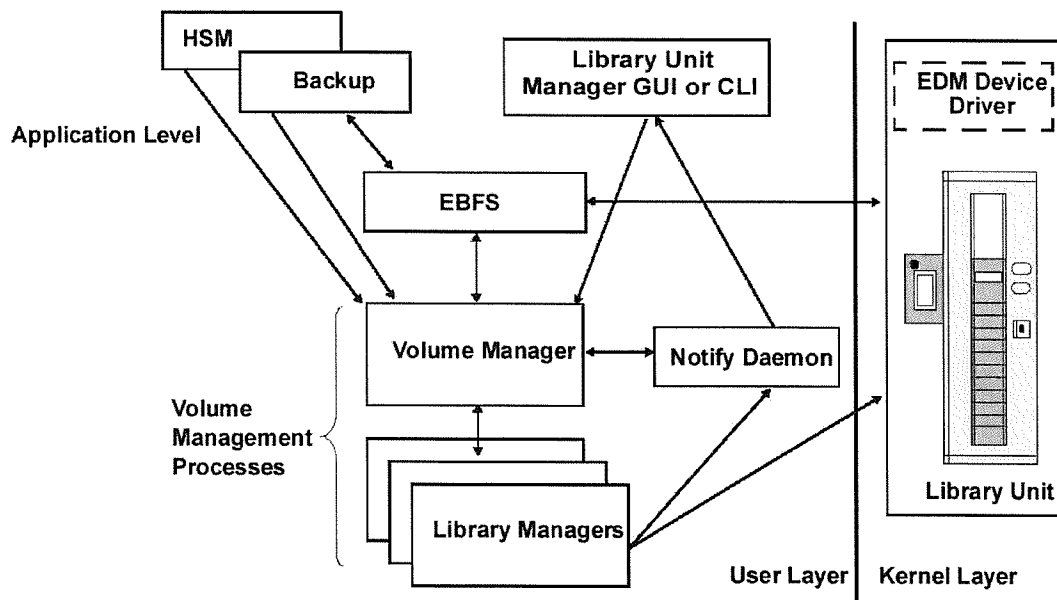
Note: To become an rvmoper member, contact your UNIX system administrator, who can add you to the group.

Volume Management Processes

The Volume Management software consists of several independent processes that together provide volume management services to Backup and HSM. Figure 8-1 illustrates these processes:

Figure 8-1

Volume Management Process Diagram



You can view currently running volume management processes by using the **evmlistd** command. (Refer to the **evmlistd** man page for more information.) Following is an example of currently running processes:

```
# evmlistd
root 3966 3964 0 Nov 18 ? 4:36 notd -d
root 3983 3964 0 Nov 18 ? 40:01 ../atl_3264_0/lmd .
root 3964 1 1 Nov 18 pts/6 10:18 /usr/epoch/bin/vmda
root 3984 3964 1 Nov 18 ? 7:05 ../offsite_0/lmd -c
root 3985 3964 1 Nov 18 ? 6:52 ../offline_0/lmd -c
Done
```

Volume Manager

The Volume Manager (vmdaemon) is the principal process in volume management. It interacts with EBFS (Epoch Bitfile System), device-specific Library Managers, and the notify daemon. EBFS enables applications, such as Backup and HSM, to write data to removable media. Library Manager daemons (lmd) control library unit operations such as robot movement which transports cartridges to and from an LU's inlet, internal drives, and storage slots. The notify daemon communicates changes between the Volume Manager and Library Managers, and EDM Library Unit Manager graphical user interface (GUI).

You can also view the current status of the Volume Manager by using the **evmstat** command. Various options enable you to view the status of components such as drives, inlets, individual library units, etc. (Refer to the **evmstat** man page for more information.)

Backup and HSM invoke requests to the Volume Manager to open and close volumes, obtain volume status, and to allocate and deallocate volumes.

Library Managers

Individual Library Manager processes manage library unit operations on a per-device basis. A Library Manager is set up for each library unit that is configured for the EDM by using the **lmconfig** utility.

Requests for media are sent from an application to the Volume Manager. Upon receipt of a request, the Volume Manager passes the request to the appropriate Library Manager for processing.

The Library Manager notifies the Volume Manager and EDM GUI, via the notify daemon, when it completes an operation.

Notify Daemon

The Volume Manager and each Library Manager communicates with the GUI by way of the notify daemon (notd). The notify daemon enables the GUI to display up-to-date status and information during system operation. For example, when you label a new piece of media, the Volume Manager sends the new volume label information to the EDM Library Unit Manager by way of the notify daemon. The Media List is updated to display information for the newly labeled volume such as its volume sequence number, the application making the request, and the trail name.

Volume Management Startup

The vmdaemon is started from the system startup file. The vmdaemon starts the notify daemon, sets the parameters defined in its configuration file (vm.cfg), and starts an lmd process for each Library Manager configured for the server.

During startup, each Library Manager daemon reads a unique configuration file (lm.cfg). The configuration file defines the name of the Library Manager, sets the hardware address of the library unit and drives, configures the number of drives and drive features, and sets the library unit's operating and

scheduling parameters. (See “Library Manager Configuration Files” on page C-9 for more information about the parameters in the Library Manager configuration file.)

Manually Stopping and Restarting the vmdaemon

Under normal operating conditions, you do not have to start up or shut down the volume management system. When you reboot the system, volume management processes are automatically shut down and restarted.

However, you may need to shut down and restart volume management if the vmdaemon fails or you determine that one or more processes are in an unknown state. Volume management does not automatically recover from unexpected vmdaemon failures.

Note: Manual shutdown of the vmdaemon should be done only by EMC field service personnel or when instructed by an EMC customer service representative.

Using **edmproc -restart**

To recover from this type of failure, use the **edmproc -restart** command. This command shuts down the remaining EDM processes and then starts up all of the processes again in the correct order. (Refer to the **edmproc** man page for more information about this command.)

You should be sure no backup, media duplication, HSM, or restore processes are running when you use this command. Use the command **vmdupd -L** to check on media duplication processes, **ebbackup -L** to check on backup processes, **ebrestore** for restore processes, or **emsstat** for HSM processes. (Refer to the appropriate man page for more information about each of these commands.)

Following is an example of the output that appears at the CLI when you use the **edmproc -restart** command:


```
# edmproc -restart
```

```
EDM daemon shutdown ...
```

```
    Shutting down System Monitoring ... Done
```

```
    Shutting down Backup Activity Monitor ... Done
```

```
    Shutting down Client daemons ... Done
```

```
    Shutting down Backup Server ... 12/27/99 15:10:25
```

```
[16226:ebcatalogd] Halt signal sent to ebcatalogd process #1830
```

```
Halt signal sent to EpochBackup Listener process #1806
```

```
Done
```

```
    Shutting down Bitfile Services ...    halting ebfsd, process id 356
```

```
    Bitfile Services shutdown complete
```

```
    Media Duplication shutdown started
```

```
    Media Duplication shutdown complete
```

```
Done
```

```
Status of file /kernel/drv/st.conf is good
```

```
    Shutting down Volume Management ... Done
```

```
    Shutting down SNMP support ... Done
```

```
EDM daemon shutdown complete
```

```
EDM daemon startup ...
```

```
    Starting SNMP support ... Done
```

```
Status of file /kernel/drv/st.conf is good
```

```
    Starting Volume Management ... Done
```

```
    Starting Bitfile Services ... Done
```

```
    Starting Backup Server ... Done
```

```
    Startup Client daemons ... Done
```

```
    Starting Backup Activity Monitor ... Done
```

```
    Starting System Monitoring ...Done
```

```
Done
```

```
EDM daemon startup complete
```

**If an Error Occurs While Using
edmproc -restart**

If the ebfsd or vmdupd processes do not shut down within one and one-half minutes of the restart request, **edmproc -restart** halts the processes and forces a shutdown. A series of error messages may appear before the shutdown that address one of the following processes: ebfsd, vmdupd, or vmdupmedia (which vmdupd controls).

For example, if a problem occurs with shutting down the ebfsd daemon, the following message appears:

```
Sending kill signal to ebfsd, process id process id
```

If more time passes without a successful shutdown, another message appears:

```
Process ebfsd pid pid is slow to shut down, sending kill signal
```

If all subsequent attempts to shut down the process are unsuccessful:

```
Unable to shut down (ebfsd) process id process id.  
You must perform a UNIX shutdown to terminate this  
process.
```

You should then reboot the EDM server to clear this process successfully.

Library Unit Operations

Library Manager daemons handle the following library unit operations:

- Inserting Media into Library Units
- Mounting and Dismounting Volumes
- Ejecting Media from a Library Unit
- Drive Scheduling and Preemption
- Library Unit Inventories

Inserting Media into Library Units

Media cartridges enter a library unit (LU) through the library unit's inlet. Library units have one of two inlet types: automatic or manual. If a library unit has an automatic inlet, the Library Manager polls the inlet periodically for incoming cartridges. If the LU has a manual inlet, you must inform the Library Manager when you place media into the inlet. You do this in the Utilities tab of the EDM GUI's Library Unit Manager window. (Refer to EDM online Help for more information about this window.)

When a Library Manager detects media in an inlet, it first checks the library unit for an available slot. If a slot is available, the robot moves it from the inlet into the next available slot of the LU. This slot becomes the volume's "home slot." The Library Manager inventories the volume and sends information for the new volume to the Volume Manager (by way of the notify daemon). The Volume Manager creates an entry in the volume catalog and notifies the EDM Library Unit Manager to display the new volume in the Media List.

Importing a Volume

Volumes are imported into the EDM system when their status is Uncataloged (refer to Chapter 7 for more information about this volume state). You must inject an uncataloged volume into a library unit before you import it into an EDM.

The import feature is generally used for restore or disaster-recovery purposes, so that you can transfer one or more volumes from one EDM to another. The receiving EDM can then obtain the same information about the volume that the original EDM had.

Refer to Chapter 19 “Recovering a Server from a Disk Failure” and Chapter 20 “Recovering a UNIX Client from Disk Failure” of this manual for information about disaster recovery.

You can import the volume into an EDM through the Library Unit Manager window of the EDM GUI (refer to online Help for instructions). You can also import a volume at the CLI by entering the command **evmimport** (refer to the `evmimport` man page for more information). For example:

```
# evmimport -V -l atl_452_0 -s 39
```

Using this command adds information about the volume to the volume catalog, and the volume is cataloged.

Importing a Duplicate Volume Before Its Original

When importing a duplicate volume into an EDM system where the original volume is unknown, the duplicate's barcode, volume ID, and sequence number are imported with it. The volume ID of the duplicate's original volume is also imported. A "placeholder" sequence number is created for the original volume in the LU. (This placeholder provides the original volume a valid sequence number; it does not affect backup or restore processes.) The original volume's barcode remains blank.

If you then import the original volume into the same EDM system, the original volume's proper sequence number replaces the placeholder sequence number that was created for it. The original volume's barcode also updates to match the real volume.

Gathering Media Information

When a volume is imported, the fields contained on the label (such as volume ID) are set in the Library Unit Manager of the EDM GUI, or by running **evmimport**. Other media information that EDM uses is set by running **ebimport** (such as the ebfs ID and ebfs directory ID).

Other information such as the amount of data written to the media (which appears in the media list information in the Library Unit manager window of the EDM GUI) cannot be retrieved. However, this lack of information does not affect normal EDM operations (backups, restores, duplications, etc.).

Inserting Cleaning Cartridges into Library Units

You insert a cleaning cartridge into a library unit (LU) as you do a data tape. The Library Manager recognizes the tape as a cleaner by its barcode when the cleaner enters the LU. (During configuration of the LU, the default barcode for cleaners is set to CLNXXX. Refer to Chapter 17 "Configuring Library Managers" for more information.)

When you inject a cleaner into an LU for the first time, the default for the maximum number of times the cleaner can be used is set to 20. As a cleaning cartridge is used, its remaining uses count is decremented.

Note: If barcodes are not used, the cleaner barcode is not CLNXXX, or cleaner barcode recognition is disabled, you must inject the cleaner through the Utilities tab of the Library Unit Manager window, or by using the **evminject -c** command at the CLI (refer to the **evminject** man page for more information).

You can change the maximum uses count by using the **evmchvol** command in which you specify the LU name and slot number where the cleaner resides, as shown:

```
# evmchvol -l library unit name -s slot number maxuses=n
```

If the cleaner was already used a number of times before being inserted into the LU, you can ensure this usage count by using the **evmchvol** command as follows:

```
# evmchvol -l library unit name -s slot number uses=n
```

You can verify the uses or maximum uses count that you set by viewing the values in the Information tab of the Library Unit Manager window, in the EDM GUI.

After the cleaner is used for the first time, the Library Manager tracks the number of times the cleaner is used until it reaches the set maximum.

Note: Be sure to have another cleaning cartridge available when a cleaner in use reaches its maximum usage.

(Refer to the **evmchvol** man page for more information about this command.)

When Drives are Busy

When you insert (inject) media into a library unit, and all drives are busy, the inject does not complete until a drive is available. An inject operation does not preempt any job that is currently using drives, even if that job is a lower priority than the job that requested the inject. (An inject requires a drive to read the volume's label.)

For example, backups have a higher priority than media duplication. But if all drives are busy with a media duplication operation and a request arrives causing the Media Requests window to open, you can insert a volume but the inject does not complete until a drive is available. (See "Drive Scheduling

and Preemption” on page 8-16 for related information.) To avoid this problem, make sure you always have sufficient media in the library unit for the higher priority job.

Inserting media for a higher priority job waits for a lower priority job to release the drive, it does not preempt any job that is currently using the drives, even if that job is of a lower priority than the job that requested the inject. (An inject requires a drive to read the volume's label.)

Note: To avoid this problem, you must make sure you have sufficient media in the library unit for the higher priority job prior to starting the job. You can also configure a trail to use fewer drives during processing (refer to EDM Online Help for more information).

Mounting and Dismounting Volumes

Library Managers handle mount and dismount requests on a priority basis. The following sections describe mounting and dismounting media.

Mounting Volumes

When a mount request arrives, the Library Manager first determines if the volume is mounted in a drive or one of the storage slots. If the volume is already mounted in a drive, the Library Manager sends the Volume Manager the drive number in which the volume is mounted.

If the volume is in one of the library unit's storage slots, the Library Manager schedules the volume for mounting. When a drive becomes available, the Library Manager mounts the volume, reads its volume label, and sends the drive number to the Volume Manager.

If the volume is not in a drive or library unit, the volume is either offline or offsite. If the volume is offline, the Volume Manager generates a request for the GUI to open the Media Request window. The Media Request window displays the volume sequence number and/or barcode ID of the volume

(and its duplicate if one exists). To respond to the mount request, the operator physically locates the volume and inserts it into the specified library unit.

After the volume is inserted into a library unit, the Library Manager schedules it for mounting and the request is removed from the Media Request window. The Library Manager mounts the volume and notifies the application (via the Volume Manager) that the volume is ready for access.

Dismounting Volumes

When a dismount request is received from an application, the Library Manager first acknowledges the request, places the volume in a dismount queue, and leaves it in the drive for a specified number of seconds (LM_MAX_IDLE_TIME determines this number, as configured in the lm.cfg file).

Note: The LM_MAX_IDLE_TIME parameter does not apply when dismounting a volume from the GUI or CLI; in either case, the volume dismounts immediately.

When the time that is specified in lm.cfg elapses, the Library Manager dismounts the volume from the drive and returns it to its home slot. If a request comes in for the same volume during this period of time, the volume is already mounted in the drive and ready for access by the application. The Library Manager avoids remounting the same volume and thereby optimizes drive access.

If the Library Manager receives a mount request for a new volume during a pending dismount and no other drive is available, the volume with the pending dismount status is immediately dismounted to free up the drive so that a new volume can be mounted.

Ejecting Media from a Library Unit

When you eject media from a library unit, the Library Manager schedules the volume(s) for removal from the library unit. Information about each ejected volume moves into the offline Library Manager. If you specify to eject the media to offsite, information moves briefly into the offline LM and then to the offsite LM.

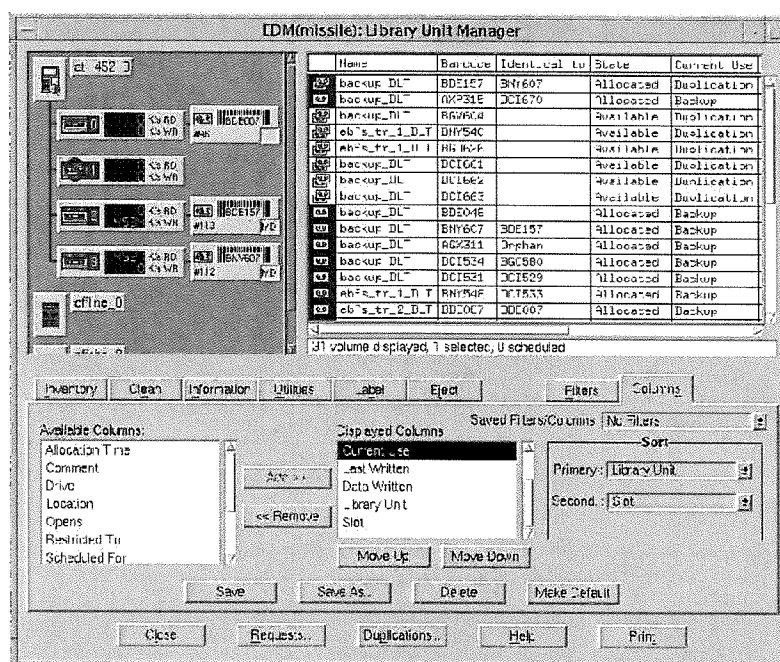
When the Volume Manager receives an eject request, it first checks the catalog for the volume's location and sends a request to the appropriate Library Manager for processing. The Library Manager, upon receipt, schedules the eject request.

If the volume is in a drive at the time the eject request is processed, the Library Manager waits for receipt of a dismount request before ejecting the volume from the library unit.

After the Library Manager ejects a volume, all outstanding requests for the volume are canceled. The Library Manager sends notification to the Volume Manager to close the request; the EDM Library Unit Manager window then reflects the change.

Ejecting Media Through the EDM GUI

You can eject media through the Library Unit Manager window (Eject tab) of the EDM GUI. In this window, select the Eject tab, as shown below. Refer to EDM Online Help for instructions on using this tab.



Ejecting Media at the CLI

At the CLI, use the following command to eject media from a library unit:

```
# evmeject
```

When you issue the command, the prompt does not reappear until the eject operation has completed. However, if you want to run **evmeject** as a background operation, use the command as follows:

```
# evmeject &
```

Refer to the **evmeject** man page for more information.

Drive Scheduling and Preemption

Each Library Manager handles drive scheduling based on a priority that the application establishes. When a Library Manager receives a request (for example, a mount request), the Library Manager adds the request to a prioritized work queue. When a drive becomes available, the Library Manager services the next work order with the highest priority.

Drive Preemption

Drive preemption occurs when a volume is mounted in a drive and an application makes a mount request for a volume with a higher priority. (The Library Manager determines preemption of a volume based on the volume's priority in the queue.) If no other drives are available, the volume with the lower priority is dismounted, which makes the drive available to the volume with the higher priority.

Verifying Priority in the Queue

An application, with a mounted and open volume, periodically polls the Volume Manager to verify whether the volume should be removed from the drive. The Volume Manager, in turn, asks the Library Manager to check for mounts of volumes with a higher priority. If a volume with a higher priority is waiting to be mounted, the Library Manager notifies the application by way of the Volume Manager to close the volume in the drive. After the application closes the drive, the Library Manager dismounts the volume, making the drive available to the volume with higher priority.

If all application requests are of equal priority, they are scheduled on a round-robin basis. For example, if five volumes have the same priority and only one drive is available, each application gets a time slice of the drive.

**LM_MAX_RESIDENT_TIME and
LM_MIN_RESIDENT_TIME**

Two parameters essentially govern drive usage:

LM_MAX_RESIDENT_TIME and LM_MIN_RESIDENT_TIME.

- LM_MAX_RESIDENT_TIME is the maximum time (in seconds) that a volume can remain in a drive before allowing preemption by a volume of the same priority. The default value for tape drives is 7200 seconds (two hours); the default value for EO drives is 120 seconds (two minutes).
- LM_MIN_RESIDENT_TIME is the minimum time (in seconds) that a volume with a lower priority can be in a drive before allowing preemption for a volume of a higher priority. The default value for tape drives is 120 seconds (two minutes); the default value for EO drives is 30 seconds.

Note: You can modify both parameters in the lm.cfg file. Refer to Appendix C "Volume Management Configuration Files" for more information about lm.cfg.

Note: Though you can set LM_MIN_RESIDENT_TIME to any value, the EDM applications (backup, restore, HSM, and media duplication) poll drive usage every five minutes. Therefore, if you set the value to less than five minutes, preemption does not occur until five minutes pass and the application verifies whether preemption is necessary.

Simultaneous Backup Example

The following example describes the process of two simultaneous backups that are contending for the same drive.

1. The first backup process sends a request to the Volume Manager to open and mount volume sequence number 42.
2. The Volume Manager checks the volume location in the volume catalog and sends a request to the appropriate Library Manager to mount volume sequence number 42.
3. The Library Manager creates a work order to mount volume sequence number 42.

4. A second backup process sends a request to the Volume Manager to open and mount volume sequence number 123.
5. The Volume Manager checks the volume location in the volume catalog and sends a request to the appropriate Library Manager to mount volume sequence number 123.
6. The Library Manager creates another work order to mount volume sequence number 123.
7. Periodically, the Library Manager is asked if the volume in the drive should be dismounted.
8. The Library Manager makes a decision based on a volume's priority. If volume sequence number 123 has a higher priority, the Library Manager removes volume sequence number 42 from the drive and mounts volume sequence number 123 (if volume number 42 was in the drive at least for the time specified in LM_MIN_RESIDENT_TIME).

If both volumes have equal priority, the drive is shared until one of the backup processes finish, or until the time specified in LM_MAX_RESIDENT_TIME has passed.

Library Unit Inventories

A Library Manager inventories the contents of a library unit based on the type and verification criteria you select from the Inventory tab of the GUI. Inventory type enables you to inventory the entire contents (all slots) or only a selected portion of a library unit.

Note: A label and barcode inventory is recommended.

For most library units, a Library Manager inventories each volume that enters a library unit by way of the inlet. If your library unit does not have an inlet, you need to update the Library Manager anytime you change the contents of the library unit by running an inventory. You must also inventory a library unit anytime you bypass normal Library Manager operations (for example, you move a volume using the library unit's front

panel switches). The type of inventory you do depends on the activity or change you make inside the library unit after opening the door.

If you open a library unit door, manually move media, and then shut the door, the status of the volumes in the library unit becomes unknown. To recover from this state, you need to run a label and barcode inventory so that the Library Manager knows what volume is in each slot. The type of inventory you perform is based on what changes (if any) you make inside the library unit after opening the door.

Note: If your library unit has an inlet, lock the front access door and always use the inlet to insert and remove tape cartridges. If you need to open the access door to insert and remove large amounts of media, be sure to perform a delta inventory after each move.

Inventory Tables

Each Library Manager maintains an internal inventory table in the appropriate Library Manager subdirectory in `/usr/epoch/etc/lm/volid.dat`.

Note: The `volid.dat` inventory file is created when a Library Manager is initially started by the `vmdaemon`. This file is used exclusively by the Library Manager and must not be deleted.

The table includes a volume ID, barcode label (if configured in `lm.cfg`), slot number, and drive location for each volume in the library unit. During an inventory, the Library Manager compares the slot contents of the library unit with the information in the inventory table. If any discrepancies are found, the Library Manager updates its table and sends the changes to the Volume Manager for cataloging.

How Inventories are Done

When a Library Manager receives a request for an inventory, it begins the process by notifying the EDM Library Unit Manager that an inventory is in progress. The word “inventory” appears next to the appropriate library unit in the Library Unit and Drives area of the GUI. The Library Manager creates a list of slots (based on the type you select) to inventory.

As the Library Manager inventories each volume in its list, the slot contents are updated in the inventory table. If it detects any changes, the Library Manager sends the slot contents to the Volume Manager for cataloging.

You can schedule an inventory during normal system operations. An inventory always has a low priority to ensure that maximum system performance is maintained. The Library Manager processes incoming requests of a higher priority, such as mount and dismount requests, before handling an inventory. If a mount request comes in for a volume that is already in the inventory queue, the Library Manager inventories the volume and removes it from the queue.

After the Library Manager inventories the last item in the list, it informs the Volume Manager and the notify daemon that it completed the inventory. The Volume Manager modifies the volume catalog based on the changes that the Library Manager provided and the GUI is updated.

During the inventory process, the Library Manager processes incoming operations that are of a higher priority. Therefore, a full library unit inventory by label does not have a noticeable effect on overall system performance.

Delta Inventory

When you perform a delta inventory, the Library Manager:

- checks each slot in the library unit to determine which slots changed.
 - If the slot was full and is now empty, the Library Manager marks the slot as empty and removes the volume from its inventory table.
 - If the slot was empty and is now full, the Library Manager marks the slot as needing verification.
 - If the barcode of a volume in a slot is different from the barcode of a volume that was previously in the slot, the Library Manager marks the slot as needing verification.
 - If no change was made to the slot, the Library Manager skips the slot.
- creates a work order for each slot that needs verification and inventories each item using the specified verification criteria.

Barcode Inventories

If a library unit is equipped and configured (in `lm.cfg`) to read barcode labels, you can verify only the barcode label or you can verify both the barcode label and the volume label.

The Library Manager does a barcode inventory by scanning the barcode label of each volume in the inventory queue. Because no volume mounting is involved, a barcode inventory takes significantly less time to complete than a label inventory. The Library Manager updates the barcode ID for each volume in its inventory table and informs the Volume Manager of any changes.

When you select Verify Both Label and Barcode from the Inventory tab of the EDM Library Unit Manager window, both the barcode label and volume label are verified for the selected inventory type. (Refer to Help for the Inventory tab for details.)

Note: It is recommended that, when volumes are regularly rearranged in a library unit through a mechanism other than the documented inject and eject operations (for example, insertion and removal through the library unit's mass-load door), you run a complete label and barcode inventory rather than the simple barcode inventory. This averts volume barcode and ID mismatch and other related problems that could arise in the volume catalog.

Note: Do not use duplicate barcodes. Attempting to add duplicate barcoded media to the system causes unpredictable results.

Cleaning Tape Drives

The EDM software detects when drives need to be cleaned. If a cleaning cartridge is loaded in the library unit, EDM mounts the cleaning cartridge and cleans the drive automatically.

The **evmclean** command (which you can add to a cron procedure) requests cleaning for specific drives in a library unit. The drives are cleaned when they become available.

evmclean cleans as many of the indicated drives as possible before exhausting the uses that remain on the cleaning cartridge. (Refer to the **evmclean** man page for more information.)

Note: For procedures about cleaning drives manually in the EDM GUI, refer to EDM Online Help.

At least one cleaning cartridge with remaining uses should always be available in each library unit. If a drive requires cleaning and a cleaning cartridge with remaining uses is not available in the library unit, the drive is disabled until it is cleaned. Any data tape that is mounted in the drive when it goes dirty is dismounted. If, at that time, no unused drives are available in the library unit, the performance of any active backup, duplication, restore or HSM operation may be negatively impacted.

Volume Allocation and Deallocation

The process of volume allocation and deallocation is initiated at the application level. When an application needs a volume allocated to a trail, it sends a request to the Volume Manager.

An application determines when data on a volume is no longer needed and is ready for deallocation. Once a volume is deallocated, it becomes available for allocation. Backup determines if a volume is ready for deallocation when expiring backups. HSM handles volume deallocation following compaction.

How Volumes are Allocated

During a backup, when the current volume is filled, the application makes a request for another volume in the same trail. The Volume Manager searches its drives and library units for an available volume that matches the request. If one is available, it is allocated to the application.

Additional volumes become available to a trail when you label media. You specify the trail name that can use it by choosing the appropriate volume template. You can also specify whether the volume is part of an available pool of new volumes, or it is available for a specific trail name.

If no available volumes are in the library unit, the Media Request window alerts the operator to make a new volume available. The window includes any prelabeled volumes that are offline and any unlabeled volumes that can be labeled for the request.

Volume Allocation Request

When an application needs a new volume, it sends a volume allocation request to the Volume Manager. The request includes a template that defines the type of volume it needs.

Upon receipt of the request, the Volume Manager creates an entry in the volume catalog, assigns a volume ID to the request, and sends the volume ID to the application. The application retains the volume ID. No physical media is associated with the request at this time; therefore, a volume sequence number is zero or barcode ID does not yet exist for the volume.

When the application is ready to use the volume, it sends an open request and a mount request with the volume ID to the Volume Manager. The Volume Manager searches the volume catalog for a suitable volume. When found, it sends a mount request to the appropriate Library Manager.

Mount Request

When the Library Manager receives the mount request, it schedules the request, mounts the volume when a drive is available, and relabels the volume as Allocated. The Library Manager notifies the Volume Manager when the volume is ready for the application to access the volume.

Note: If no available volumes match the request, the Volume Manager sends a volume allocation request notification to the EDM Library Unit Manager. The Media Request window opens and displays NEW in the Sequence # field.

Volume Use

The application retains the volume ID and continues to use the volume until it completes writing to the volume or the volume fills up (for example, a tape reaches the end of the media). When the application fills one volume and needs another, it sends another volume allocation request to the Volume Manager and the process repeats.

When the Volume Manager looks for a suitable volume, it looks for a volume that meets these requirements:

- The volume is not currently allocated to another application.
- The media type matches the media type that is specified in the template.
- The volume did not exceed its maximum use count. If the maximum use count is set to 0, its use is unlimited.
- The volume restriction is reviewed (volume restrictions for a given media set are defined when the backup is configured):
 - The volume is labeled as Unrestricted, which allows the volume to be allocated to *any* requesting application.
 - The volume is labeled as Restricted by Application which means the volume can be allocated only to the specified application.
 - The volume is Restricted by Name which means the volume can be allocated only if the trail name is the same as that specified by the requesting application. This ensures that once a volume is allocated to a specific trail, it cannot be reallocated to a different trail.

Duplicate Volume Sequence Numbers

Normally, volume management does not allow two volumes of the same media type to share the same volume sequence number. If you attempt to import a volume with a sequence number that already exists on the server, you are prompted to either cancel the import operation or to delete the existing volume that has the same sequence number.

However, if your site has existing volumes with duplicate sequence numbers and you want to import both (or all) volumes, you can override this restriction. The imported volumes then have the same sequence numbers but different barcodes and volume IDs.

Using duplicate numbers does not affect the running system. The sequence numbers enable you to identify individual volumes and do not affect the system's operation.

To allow importing of duplicate sequence numbers, change the value of `VM_ALLOW_DUP_SEQ_IMPORT` (in the file `/usr/epoch/etc/vm/vm.cfg`) to "yes."

Note: After you change this value to "yes," no warning is given when you import volumes with duplicate sequence numbers.

Following are the steps to incorporate changes to the `vm.cfg` file after adding support to allow duplicate sequence numbers:

1. Obtain the process ID (pid) of the `vmdaemon`:

```
# evmlistd

root    382      1  0 12:00:29 ?          2:26
/usr/epoch/bin/vmdaemon -d
```

2. Send a HUP signal to the `vmdaemon` to pick up any changes to the `vm.cfg` file:

```
# kill -HUP vmdaemon_pid
```

For example:

```
# kill -HUP 382
```

When Volumes are Deallocated

When an application deallocates a volume, the Volume Manager takes action based on the volume's media type.

- If the media is DLT, HITC, or DTF, the volume state changes to Available. When a tape cartridge reaches its maximum usage, the volume state changes to Expired.
- If the media is EO, and is configured for auto-erase, the Volume Manager erases the volume and changes its state to Available.

You can relabel a two-sided optical disk only when both sides become available. The Volume Manager treats all other state changes on an individual side basis. Thus, one side of the disk can be allocated without requiring that the other side be allocated; one side of the optical disk can be expired, erased, made available, and allocated without affecting the other side.

- If the media is a WORM (Write Once, Read Many) optical disk, its data cannot be erased. The state of a WORM optical disk starts as unlabeled. After it is labeled, the disk has a life cycle of Available, Allocated, and then Expired.

For more information about volume states, see "Volume Life Cycle" on page 7-4.

9 Media Duplication

Media duplication enables you to create a duplicate set of backup media automatically after each backup session. You can then use the duplicate set for disaster recovery purposes. This chapter includes the following topics:

- The Media Duplication Process
- Starting Duplication
- Determining Duplicates of an Original Volume
- Manually Disabling and Re-enabling Duplication
- Pausing, Resuming, Canceling, or Removing a Duplication
- Restoring from Backup or Duplicate
- If a Duplication Fails
- Restoring from Backup or Duplicate
- Viewing Reports on Duplications
- Importing a Duplicate Volume
- Rejecting a Mount Request
- Expiring a Duplicate Volume

The Media Duplication Process

Media duplication enables you to make a duplicate set of tape media automatically after a regular backup session. You can then send the original volume offsite and keep the duplicate copy onsite for restore purposes. (Either an original or duplicate may be used for restore purposes.)

A duplicate set of media is of the same media type and has the same theoretical size as the original media. Duplicate media are also uniquely labeled; that is, a duplicate volume has a different volume ID than its corresponding original.

Duplication of backup media can occur automatically after each backup session (unless a trail is set for manual duplication). All media that backup creates can be scheduled for duplication.

Note: You cannot duplicate volumes that are used by Hierarchical Storage Management (HSM).

Duplication of a trail starts after its backup completes. However, if backup of a trail requires more than one volume, neither is duplicated until the backup of the entire trail completes.

Media duplication can run when an unrelated backup or restore process is running, although backup or restore processes take precedence. If a backup or restore activity starts during media duplication and only one drive is still available, both the original and duplicate in a current duplication process share that drive. If no other drives are available, duplication is suspended until an operation with higher priority completes.

Older-generation duplicate volumes are purged automatically when a trail is configured to use new mode for duplication. (Refer to “Starting Duplication” on page 9-3.) These older duplicate volumes are purged when the current duplication of the original volume completes successfully. Only one allocated duplicate volume exists for an original volume.

Note: The automatic purge feature cannot be disabled.

Media Duplication Commands

To manage and control media duplication processes, you can manage the `vmdupd`, and `vmdup` daemons, and use the **`vmdupcfg`** command. The following briefly describes each of these; examples of using each are provided later in this chapter. (Refer to the appropriate man page for detailed information.)

The `vmdupd` daemon manages media duplication. This daemon, which runs in the background, is part of the normal EDM startup process. `vmdupd` monitors online tape volumes that require duplication and starts the `vmdupmedia` processes that perform duplication of those volumes. When run from the command line, you can monitor and control `vmdupd`'s actions.

The `vmdup` daemon command manages the media duplication scheduling queue. You use this command to start or stop duplication for a trail that is configured for manual duplication. You can also remove duplications from the schedule queue before they begin duplication, or reschedule failed duplications.

Using the **`vmdupcfg`** command enables you to view the current state of duplication parameters, or modify particular parameters. This command also allows you to enable or disable media duplication system wide.

Starting Duplication

When starting media duplication, you prepare a backup volume for duplication, configure the duplication in append or new mode, configure a backup trail for manual or automatic duplication, and initiate duplication.

The following sections describe each of these procedures.

Preparing a Backup Volume for Duplication

At the beginning of the backup process, a padded tape label (a tape label and padding block) is written to an *original* volume, even if duplication is disabled. This helps to ensure that all of the original data fits on the duplicate media if the duplicate is found to have many bad blocks. Thus, the actual data that is written to the duplicate is exactly the same as that on the original volume. That is, there is always a one-to-one correspondence between the original volume and the duplicate. (The current default pad block is 10 Mb for all tape media.)

You can control the size of the padding block of an original volume at the time a volume is created. For example, you may want to change the pad size on new volumes if your volumes tend to have a high number of bad blocks. You specify the padding block size at the command line using the **vmdupcfg** command with the optional **-tape_pad** argument as follows:

```
# vmdupcfg -set -tape_pad n
```

where *n* equals the number of blocks. The value of *n* can range from 1 to 50, where 1 = 10 Mb and 50 = 500 Mb; the default value is 10.

Note: The new tape pad size takes effect the next time a backup process allocates new media.

For example, if you set the number of blocks to 4:

```
# vmdupcfg -set -tape_pad 4
```

You can check the setting with **vmdupcfg** as shown:

```
# vmdupcfg
Media duplication enabled.
No automatic media ejection from the library unit(s).
Tape pad blocks: 4
Max concurrent dups: 1
```

Selecting Append Mode or New Mode

Append Mode

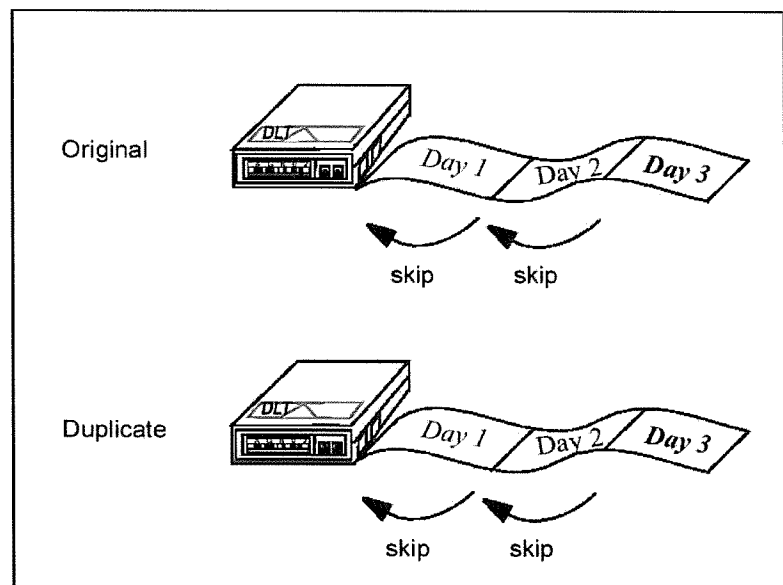
You can configure media duplication for append mode or new mode, as described below.

In append mode (the default), duplication follows the policy that the backup procedure uses. That is, if the backup appends, or adds, data to an existing volume or writes data to a new volume, duplication also appends to an existing volume or uses a new volume, respectively.

When appending to an existing duplicate volume for a trail, duplication starts from the beginning of the volume but skips the previous backup duplications that correspond to the original backup volume. It then adds new backup information to the end, as shown in Figure 9-1.

Figure 9-1

Append Mode: add data for Day 3 after Day 2.



In the above example, backup of a trail for Day 3 is appended to the original volume; the data for Day 1 and Day 2 is skipped and Day 3 is added. Duplication of this original volume follows the same policy as that for the original; data for Day 3 is appended after that of Day 2 on the duplicate volume.

New Mode

When you use new mode for duplication, a new volume is allocated for the duplication even if the backup procedure calls for appending to its existing backup volume. Using new mode duplicates the entire tape each time, from beginning to end.

In Figure 9-2, data for Days 1, 2, and 3 of a trail's backup exist on one original volume. During duplication in new mode, data for Day 1 is written on one duplicate volume, data for Days 1 and 2 is written on another, and data for Days 1, 2, and 3 is written on yet another. The duplicate volumes with data for Day 1 and Days 1 and 2 are now considered older generation volumes and are no longer valid. As each duplication completes successfully, the previous, older duplicate volume is reallocated for re-use.

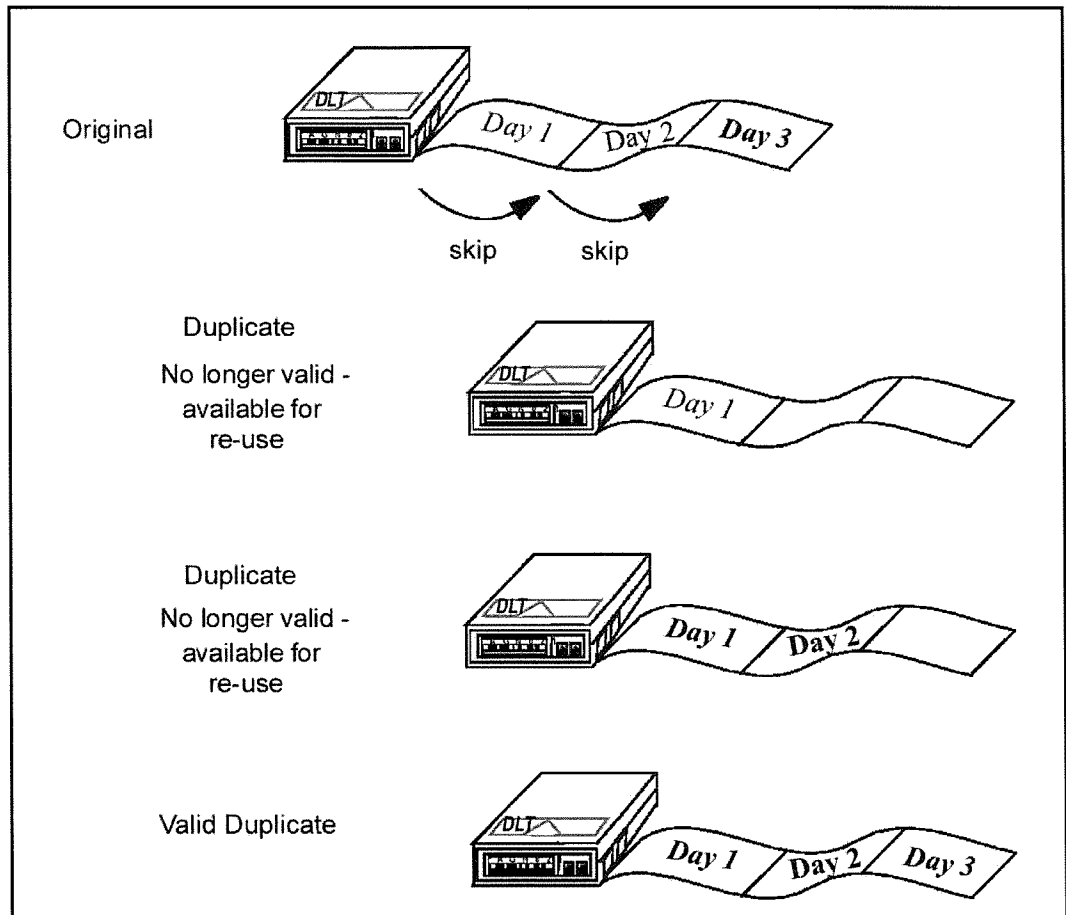
Note: It is **strongly** recommended that you use append mode for duplications. When using new mode, the time to duplicate the original volume increases as more backups are appended to it.

If it is necessary to use new mode, select the Media tab in the Backup Configuration window. In the Duplicate Media section of the window, select the button that is labeled Use New Media for Duplication.

Note: Be sure that new media is available for use in the library unit.

Figure 9-2

Using New Mode for Duplication.



Configuring a Trail for Duplication

You configure a backup trail for automatic or manual duplication in the Backup Configuration window of the EDM GUI. In this window, you select the Media tab and then click on the Duplicate Media button.

Selecting automatic duplication enables automatic duplication of a trail when a backup completes. In the mounted volume display area of the Library Unit Manager (LM) window, an application ID of "MD" on the mounted original and duplicate volumes indicates that duplication is in progress.

Note: It is recommended that you use automatic duplication (the default) when duplication is enabled.

Selecting manual duplication in the Backup Configuration window enables duplication of a trail if you wish to duplicate the media at a later time (refer to "Initiating Manual Duplication" below).

Note: Refer to "Backup Trailsets" on page B-58 for an example of a trail that is configured for duplication.

Initiating Manual Duplication at the CLI

You initiate manual duplication of an original volume by trail name at the command line by using the **vmdup** command with the **-set** and **-trail** options, and the trail that you wish to duplicate:

```
# vmdup -set -trail trailname
```

An example with output follows:

```
# vmdup -set -trail backup_DLT
```

```
Trail backup_DLT scheduled for duplication
```

You can also initiate duplication of an original volume by trail ID by using the **-trailID** option and the trail ID. To obtain a trail ID, run **ebreport media**; a trail ID appears as a rotation ID in the listing.

```
# vmdup -set -trailID trailID
```

An example with output follows:

```
# vmdup -set -trailID 82D82B05.A9730CA5.00000200.380491BA
```

```
TrailId 82D82B05.A9730CA5.00000200.380491BA
scheduled for duplication
```

Note: Ensure that media is available online for duplicate volumes.

Note: Ensure that the original volume is *online* (not offline or offsite) before initiating duplication.

You can use the command **vmdupd -all** to verify that duplication was scheduled. This command displays all active, suspended, and scheduled duplications. (See also “Verifying the Status of a Duplication” below.)

An example follows:

```
# vmdupd -all
```

```
Media duplication daemon started.
```

```
Active Duplications
```

```
Trail: backup_DLT Mode: append Type: DLT tape Status: ACTIVE
Orig Vol: A1D7F9BD71812B77 (BDE098) Seq #: 000025 in TLU: atl_3264_0
Dup Vol: 45D81F75C195EEF0 (ASV891) Seq #: 000027 in TLU: atl_452_0
40% Complete Duration: 000 Hrs. 53 Min. Start Time: 12/31/1999 13:27:16
Blocks Copied: 141381 Total Blocks to Copy: 350343
```

```
No Suspended Duplications found
```

```
No Scheduled Duplications found
```

Note: A completed duplication may still appear as Active with a completion status of 100% when you check its status by using **vmdupd -all**. This duplication is still transitioning from an Active state to the duplication state of Done.

Determining Duplicates of an Original Volume

In the Library Unit Manager window of the EDM GUI, you can determine whether a duplicate exists for an original backup volume. In the media list (as shown below), you can identify a duplicate in a number of ways:

- The media list icon in the first column indicates a duplicate by a double tape graphic
- "Duplication" appears in the Current Use column if the volume is a duplicate
- In the Identical To column, the barcode for a duplicate volume appears with its original, and vice versa

Note: Refer to the Library Manager Online Help (Columns tab) for instructions on adding columns to the media list.

	Name	Barcode	Identical to	Current Use	Volume ID
	backup_DLT	BNY540	BNY548	Backup	ACE2420CF8F8504C
	backup_DLT	BDE087		Backup	FBE13ABEBF76E197
	backup_DLT	BG0626	DCI534	Duplication	8FE145CACA372FD1
	daily_DLT	COM044	COM261	Duplication	34E22200A9729F24
	backup_DLT	DCI534	BG0626	Backup	B8E144B5903F3A34
	backup_DLT	DCI532		Backup	5CE114F473735BC3
	backup_DLT	BG0581	BNY612	Backup	DEE1CE64363DF008
	daily_DLT	COM261	BG0624	Backup	2EE06992909077EE
	backup_DLT	AXP387		Backup	A2E17940EFD119E9
	backup_DLT	BNY612	BG0581	Duplication	7DE21760EBCB55F4
	daily_DLT	BG0624	COM261	Duplication	48E1E957CFDFC4C3
	backup_DLT	DCI531		Backup	F4E15B2F2D2C2E0F
	backup_DLT	BNY548	BNY540	Duplication	64E24220B15F74BE
	backup_DLT	DCI533		Backup	07E11511947F266E
	DLT	BNY607			FBE2373933BD07FF

24 volume displayed, 0 selected, 0 scheduled

You can determine the duplicate of an original at the CLI by using the **ebreport media** command. This command generates a report that lists currently allocated volumes. Refer to "Viewing Reports on Duplications" on page 9-25, and the **ebreport** man page, for more information about this command.

Note: If more than one library unit of a given media type is attached to your EDM, you cannot tell in which LU your duplicate media will reside.

You can also use the **evmstat -c** command to view a list of all catalogued volumes (see the example below). The entries in this list identify original and duplicate volumes, and the original volume for each duplicate. (Refer to the **evmstat** man page for more information about this command.)

* Mtype	Seq/BC	Name	LU Name	Volume Type	Vol_ID	Original Vol_ID	Generation
C DLT	BNY574	backup_DLT	offline_0	none	5ED19D47EED28D1C		0
C DLT	BDE133	backup_DLT	offline_0	duplicate	48D19D5CCFBF3754	CAD19D501B028C8E	0
C DLT	BDE145	backup_DLT	ex_210_0	original	CAD19D501B028C8E		0

Setting the Maximum Number of Concurrent Duplications

The system is configured at startup to run no more than one duplication at a time. This is based on the assumption that media duplication has at least two drives available for its use. However, you can configure multiple duplications to run concurrently if four or more drives are available. You configure concurrent duplications in the Backup Configuration window of the EDM GUI, or at the command line.

The number of duplications that you can set is based on the total number of drives that media duplication can use. For every pair of available drives, you can increase the number of duplications by 1. For example, with six available drives you can set this number to any value from 1 to 3.

Configuring Concurrent Duplications in the EDM GUI

In the Backup Configuration window, select the Server tab. In the Duplicate Media section, set the Maximum Concurrent Duplications from 1 to 10; the default is 1.

Note: Do not set the maximum number of duplications to a value that is greater than half the number of drives. Otherwise, *significant* performance degradation to the duplications occurs as the duplications must then share the drives among them.

Configuring Concurrent Duplications at the CLI

At the command line, set the number of concurrent duplications by using the **vmdupcfg** command as follows:

```
# vmdupcfg -set -max_dups N
```

where *N* is the number of duplications, from 1 to 10.

For example:

```
# vmdupcfg -set -max_dups 4
```

Confirm your change as follows:

```
# vmdupcfg
Media duplication enabled.
No automatic media ejection from the library unit(s).
Tape pad blocks: 1
Max concurrent dups: 4
```

If two separate library units (LUs) are being used in your system, set *N* based on the number of drives in the system that are available for duplication. For example, in a system with a two-drive LU and a six-drive LU, you can set *N* to 4 if either or both are used for duplication.

Note: If you set the maximum number of duplications at a value that is greater than the number of available drives in the system, a warning message appears. The message contains the total number of drives for a given drive type and the drive type name; for example:

```
WARNING - The server is configured with
10 DLT7000 drives. The max_dups
parameter will be updated to support 6
duplications. This drive configuration
will only support 5 duplications.
```

Verifying the Status of a Duplication

You can verify the status of active, suspended, scheduled, and failed duplications at the command line by using the **vmdupd** command. (Refer to the **vmdupd** man page for more information.)

- **vmdupd -all** displays all active, suspended, scheduled, and failed duplications, as shown in the example below. For each duplication status, the following information appears:
 - the trail that is being backed up
 - duplication mode (new or append)
 - volume type
 - duplication status (ACTIVE, SUSPENDED, CANCELED, SCHEDULED, or FAILED)
 - original volume ID
 - sequence number
 - duplicate volume ID
 - total number of blocks that are to be duplicated

(Refer to “Viewing Reports on Duplications” on page 9-25 for more information about duplication status.)

```
# vmdupd -all
```

```
Media duplication daemon started.
```

```
No Active Duplications found
```

```
No Suspended Duplications found
```

```
No Scheduled Duplications found
```

```
Failed Duplications
```

```
Trail: backup_DLT Mode: append Type: DLT tape Status: CANCELED
```

```
Orig Vol: FBD79BDAD25C1C62 (BDE099) Seq #: 000005 in TLU: atl_3264_0
```

```
Dup Vol: None
```

```
Total Blocks to Copy: 12817
```

```
Trail: backup_DLT Mode: append Type: DLT tape Status: CANCELED
```

```
Orig Vol: A8D7969E4D438EE3 (BDE137) Seq #: 000003 in TLU: atl_3264_0
```

```
Dup Vol: None
```

```
Total Blocks to Copy: 75
```

If a Duplication Was Scheduled for an Offline Volume

If you move offline a volume that is scheduled for duplication, the duplication still appears in the output when you run **vmdupd -all**. If you want to remove the scheduled duplication, you must run **vmdupd -cancel** to designate it as Failed, and then run **vmdupd -remove** to remove the duplication from the Failed queue.

Manually Disabling and Re-enabling Duplication

Though media duplication is enabled automatically, you can manually disable and re-enable media duplication system-wide at the command line. However, disabling duplication is not recommended unless absolutely necessary (for example, if duplication of original volumes is not to be performed for any volumes in the system at any time).

Procedures for disabling and re-enabling duplication are described below.

Note: Any configuration values that you set for duplication (pad blocks, maximum number of duplications) are lost when you disable duplication. You must reset them when you re-enable duplication if you do not wish to use the default values.

Disabling Duplication

You disable duplication by halting the vmdupd daemon that controls media duplication, and then disabling duplication. Any duplications that are in progress are also suspended, but are completed when duplication is enabled and the vmdupd daemon is restarted.

Use the following procedure at the command line to disable media duplication for the entire system:

1. Run **vmdupd -halt** to halt the vmdupd daemon.
2. Verify with the following command that the daemon no longer exists:

```
# ps -aef | grep vmdupd | grep -v grep
```

The **ps** command enables you to view information about active processes. Refer to the **ps** man page for more information about this command and its options.

3. Run **vmdupcfg -reset** to disable duplication.

Re-enabling Duplication

The re-enabling process also involves two steps; you restart the vmdupd daemon and then enable duplication. Any suspended duplications are then completed.

Use the following procedure to re-enable media duplication for the entire system:

1. Run **/usr/epoch/bin/vmdupd &** to restart the vmdupd daemon.
2. Verify that the daemon started with the command:

```
# ps -aef | grep vmdupd | grep -v grep
```

An example follows:

```
client:> ps -aef | grep vmdupd | grep -v grep
```

```
root 2424 1 0 09:53:05 pts/2 1:12 /usr/epoch/bin/vm
```

3. Run **vmdupcfg -set** to enable duplication; remember that using this command resets the duplication parameters back to the default values.
4. Verify the operation by using **vmdupcfg** with no arguments:

```
# vmdupcfg
```

```
Media duplication enabled.
```

```
No automatic media ejection from the library unit(s).
```

```
Tape pad blocks: 1
```

```
Max concurrent dups: 1
```

Note: You need to reset any values (tape_pad, max_dups) that were previously set before duplication was disabled.

Pausing, Resuming, Canceling, or Removing a Duplication

You can temporarily pause duplication in progress at the command line (for example, if it is necessary to allow other operations to complete).

Note: Before pausing, verify with the **vmdupd -all** command that duplication is ACTIVE.

Pausing Duplication

To pause duplication, use the command **vmdupd -stop**. After entering **vmdupd -all** again, notice that the state changed from ACTIVE to SUSPENDED. This shuts down the duplication process and volumes are dismounted from drives. In the Library Unit Manager (LM) window, also notice that the application ID of “MD” disappears from the volumes that are being used for duplication.

Note: This command disables scheduling of all duplications system wide.

Resuming Duplication

To continue a paused duplication, use the command **vmdup -cont**. Verify with **vmdupd -all** that the state of duplication changed from SUSPENDED to ACTIVE (allow a few moments for the state to change). In the Library Unit Manager (LM) window, the volumes are remounted. Notice that the application ID of “MD” reappears on the volumes in use.

Canceling Duplication

You can cancel an active, suspended, or scheduled duplication. Canceling the process sets the duplication state to FAILED.

Canceling an active duplication shuts down the process; when the duplication is rescheduled, the entire volume is duplicated.

To cancel a duplication, you use the following command:

```
# vmdup -cancel -volID <orig volID>
```

where *orig volID* is the ID of the corresponding original volume.

The resulting output of the canceled duplication is similar to the following. The value within parentheses is the volume's barcode.

```
# vmdup -cancel -volID 7DD7E5CDDCD690DB
```

```
Duplication for volume 7DD7E5CDDCD690DB (AAC009)
canceled
```

If you wish to check on the status of the canceled duplication, you can use the **vmdupd** command. The duplication status should appear as FAILED, as shown:


```
# vmdupd -all
Media duplication daemon started.
No Active Duplications found
No Suspended Duplications found
No Scheduled Duplications found
Failed Duplications
Trail: backup_DLT Mode: append Type: DLT tape Status: FAILED
Orig Vol: 7DD7E5CDDCD690DB (AAC009) Seq #: 000004 in TLU: de_x7(
Dup Vol: None
Total Blocks to Copy: 15445
```

Removing a Failed Duplication from the Queue

You can remove a failed duplication from the queue by using the **vmdup -remove** command at the CLI. Using this command deallocates all duplicate volumes that are associated with the original volume, which makes them available for reuse.

The **vmdupd -remove** command also clears the duplication flags for the original volume, which removes the original volume from the failed duplication queue, and removes the duplication state of Failed from the original volume.

If another backup is scheduled for the original volume, the duplication of that volume is also scheduled.

Use this command as follows:

```
# vmdup -remove -volID original volume ID
```

For example,

```
# vmdup -remove -volID 5DDC2A1510B0A3D0
```

Volume 5DDC2A1510B0A3D0 removed from the failed duplication queue.

You can then use the **vmdupd -all** command to verify that this operation is successful.

If a Duplication Fails

During media duplication, the duplication is set to FAILED if:

- you reject a volume request for a specific duplicate volume at the beginning of a duplication process
- you reject a queued request for an available duplicate volume (the queued request no longer reappears)
- a request occurs for mounting and relabeling a duplicate volume that is considered "unmountable" (for example, barcode mismatch)
- you cancel a duplication (refer to "Canceling Duplication" on page 9-17)
- a read error occurs on the original volume
- a write error occurs on the duplicate volume

Be sure to reschedule the duplication as soon as possible so that duplication of the specific volume can resume after backup. Otherwise, if the trail supports automatic duplication, subsequent backups to that specific trail do not schedule duplications automatically until the next rotation.

Rescheduling a Failed Duplication

If a duplication failed for some reason, you can reschedule it in the Library Unit Manager (LM) window of the EDM GUI, or at the command line.

Note: Always check the uses count on the duplicate that is still mounted in the drive to ensure that it is less than the maximum uses for the volume. This prevents the volume status from changing to Expired for the mounted volume. (If a mounted volume's status changes to Expired, you must manually remove the volume from the drive.)

Rescheduling a Failed Duplication Through the GUI

In the LM window, click on the Duplications button. The Media Duplication Control window that appears lists all failed duplications. In addition to the Failed status, information for the failed duplication also includes the trail name and sequence and bar code numbers for the original and duplicate volumes. (Refer to “Viewing Reports on Duplications” on page 9-25 for information about rescheduling a failed duplication.)

To reschedule a duplication, select the volume for duplication in the Media Duplication Control window and then click on the Reschedule button. (Holding the Shift key while selecting enables you to select more than one volume at a time.) Information for the selected volume then disappears from the window.

(You can also access the Media Duplication Control window in the EDM Main window by selecting Duplications in the Media pull-down menu, or by selecting a Library Unit in the window, clicking on the right mouse button, and choosing Duplications from the pop-up menu.)

Note: It is important that failed duplications be rescheduled, as no subsequent backups to the original are duplicated until rescheduling occurs and duplication is successful.

Rescheduling a Failed Duplication at the CLI

When rescheduling a failed duplication at the CLI, you can view all failed duplications by using the **vmdupd -all** command as follows:

```
#vmdupd -all
Media duplication daemon started.
No Active Duplications found
No Suspended Duplications found
No Scheduled Duplications found
Failed Duplications
Trail: misk_ms_DLT Mode: append Type: DLT tape Status: FAILE
Orig Vol: E1D86D2C8219CB14 (BDE145) Seq #: 000018 in TLU: atl_
Dup Vol: None
Total Blocks to Copy: 400
```

Then enter the following:

```
# vmdup -reschedule -volID volume_ID
```

where *volume_ID* is the original's volume ID. The failed duplicate is deallocated and an available volume is selected.

Following is an example of the resulting output (the barcode, if any, appears in parentheses after the volume ID):

```
# vmdup -reschedule -volID E1D86D2C8219CB14
Volume E1D86D2C8219CB14 (BDE145) rescheduled for
duplication
```

Use the **vmdupd -all** command to verify that the duplication is scheduled:

```
# vmdupd -all
Media duplication daemon started.

No Active Duplications found
No Suspended Duplications found
Scheduled Duplications

Trail: backup_DLT Mode: append Type: DLT tape Status: SCHEDULED
Orig Vol: E1D86D2C8219CB14 (BDE145) Seq #: 000024 in TLU: atl_3264_0
Dup Vol: None
Total Blocks to Copy: 19629
```

Note: You cannot reschedule a volume that is already scheduled for duplication; otherwise, an error message appears. You must first cancel the duplication by using the command **vmdup -cancel -<volume_ID>** and then reschedule.

Rescheduling Duplication of an Offline Original Volume

If you try to reschedule duplication of an offline original volume at the CLI, a message appears that states that the original is offline. For example:

```
# vmdup -reschedule -volID <volume ID>
```

```
Cannot schedule duplication for original volume:
<volume ID> (barcode, if any), volume is in library
unit: offline_0
```

You must then inject the offline volume into the library unit before rescheduling it.

Rescheduling Duplication of a Single Volume for Archival Purposes

An occasion may arise in which, for archival purposes, you want to duplicate a volume of a trail that is not normally scheduled for manual or automatic duplication.

Before rescheduling the volume for duplication, first run **ebreport media** to find the volume ID of the volume to duplicate:

```
# ebreport media

Rotations for Template "default", Trail "backup_DLT", Primary Trailset

12/31/1999 18:03:03 Rotation ID:56D874D8.106F916B.00000200.390B91E2, 35 backups

Media duplication not used

*Orig Vol: 56D874D8106F916B (BDE132), Seq #: 000026 in TLU: at_452_0, media: DLT
Orig Vol: 13D874E52C75916D (BDE023), Seq #: 000032 in TLU: at_452_0, media: DLT
```

Next, use the **vmdup** command to reschedule duplication of the selected volume:

```
# vmdup -reschedule -volID <volume ID>
```

Following is an example of the resulting output (the barcode, if any, appears in parentheses after the volume ID):

```
# vmdup -reschedule -volID 56D874D8106F916B
```

```
Volume 56D874D8106F916B (BDE132) rescheduled for
duplication
```

Then, by using the **vmdupd -all** command you can verify that the duplication is scheduled.

```
# vmdupd -all
Media duplication daemon started.
No Active Duplications found
No Suspended Duplications found
Scheduled Duplications
Trail: backup_DLT Mode: append Type: DLT tape Status: SCHEDULED
Orig Vol: 56D874D8106F916B (BDE132) Seq #: 000024 in TLU: at_3264_0
Dup Vol: None
Total Blocks to Copy: 19629
```

Restoring from Backup or Duplicate

When you restore a backup, you can use either an original backup volume or an up-to-date duplicate volume. If the original volume is physically present in a library unit (that is, not offline or offsite), the original is automatically used for the restore.

Note: An up-to-date duplicate implies that no additional backups were appended to the original since this duplicate was completed. Therefore, the duplicate volume is an exact duplicate of the original

If the original volume is offline or offsite but an up-to-date duplicate volume is physically present in a library unit, the duplicate volume is automatically used.

If both the original and up-to-date duplicate are offline or offsite, processing suspends until appropriate media is injected into the library unit. Within the EDM GUI, the Volume Request window appears, which prompts you for the original or current duplicate. You must then load either volume into the library unit so that the restore process can use it.

Note: The duplicate volume is not substituted for the original volume during a restore if the original was modified since the last duplication.

If an Original Volume is Defective

If an original volume in your library unit is defective and a valid, current duplicate is available offline, eject the original volume so that it is no longer used for restoring files or doing additional backups.

Viewing Reports on Duplications

You can use the **ebreport media** and **ebreport duplicate** commands to check the status of duplicate media. The reports that these commands generate are described below.

(Refer to the **ebreport** man page for more information.)

ebreport media Report

The report that **ebreport media** generates reports lists all currently allocated volumes by volume ID and barcode (where applicable), and identifies whether a volume is an original or a duplicate. Also listed is whether duplication is enabled for a rotation of a trail, and if so, how many duplicates were made.

A sample report is shown:

```
# ebreport media
```

```
EDM Backup Media Report for server edm on Dec 31 23:07:40 1999
Report options: none
```

```
Summary of all media, listed by media rotation groups
```

```
Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary Trailset
```

```
12/31/1999 09:46:51 Rotation ID:A1D7F9BD.71812B77.00000200.F206F11B, 104 bac
Media duplication used on 1 copy
```

```
*Orig Vol: A1D7F9BD71812B77 (BDE098), Seq #: 000025 in TLU: at_3264_0, media: DLT ta
Dup Vol: 45D81F75C195EEF0 (ASV891), Seq #: 000027 in TLU: at_452_0, media: DLT tap
Duplication State Active, Empty, Duplication Date 12/14/1999 13:43:
Dup Vol: 40D81EE7477F8BDA (BDE146), Seq #: 000017 in TLU: at_3264_0, media: DLT ta
Duplication State Done, Successful, Duplication Date 12/31/1999 13:
```


If the trail supports duplication and no currently allocated duplicate volume is available for the original, "None" appears for duplicate volume information, as shown:

```
12/31/1999 23:57:57 Rotation ID:65D8498A.FD4DC69B.00000200.540819F4, 2 backups
      Media duplication used on 1 copy
*Orig Vol: 65D8498AFD4DC69B (BDE012), Seq #: 000020 in TLU: at_452_0, media: DLT
  Dup Vol: None

      State Done
```

Note: If you see "None" in a report, you should research the reason for the entry further. This may indicate that a duplication process did not run correctly.

For each rotation of a schedule template, just before the volumes are listed, one of the following appears on the screen:

Media duplication used on 1 copy

Media duplication not used

Note: When duplication is enabled for a trail, the report always states that duplication is used.

The volume state follows the word "State" and appears as Scheduled, Active, Suspended, Resuming, Failed, Done, or Imported.

The duplication status follows the volume status and appears as Empty or Old (another backup was appended to the original volume but not duplicated), depending on whether the duplicate is valid. (Refer to "Importing a Duplicate Volume" on page 9-29.)

The duplication volume states are listed in on page 9-28.

ebreport duplicate Report

The report that **ebreport duplicate** generates includes the information that the **ebreport media** report contains. In addition, the **ebreport duplicate** report also provides the mode of duplication (append or new), the total number of blocks that were duplicated, the start and stop times of the duplication, the end time of the last duplication, and the duplication expiration date. A sample report is shown:

ebreport duplicate

Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary Trailset

12/31/1999 09:46:51 Rotation ID:A1D7F9BD.71812B77.00000200.F206F11B, 104 backups
Media duplication used on 1 copy

Duplication State: Done, Old, Mode: New

*Orig Vol: A1D7F9BD71812B77 (BDE098) Seq. #: 000025 in TLU: atl_3264_0, media: DLT tape

Dup Vol: 40D81EE7477F8BDA (BDE146) Seq. #: 000017 in TLU: atl_3264_0, media: DLT tape

Total Blocks: 349028 Start Time: 12/22/1999 10:54:26 End Time: 12/22/1999 13:06:21

Duration: 001 Hrs. 31 Min., Duplicate Expiration Date: 12/25/1999

12/31/1999 12:57:57 Rotation ID:65D8498A.FD4DC69B.00000200.540819F4, 2 backups
Media duplication used on 1 copy

If the trail supports duplication and no currently allocated duplicate volume is available for the original, "None" appears for duplicate volume information, as shown below.

Note: If you see "None" in a report, you should research the reason for the entry further. This may indicate that a duplication process did not run correctly.

Duplication State: Done, Mode: Append

*Orig Vol: 65D8498AFD4DC69B (BDE012) Seq. #: 000020 in TLU: atl_452_0, media: DLT tape

Dup Vol: None

Total Blocks: 207

Table 9-1 below lists several examples of duplicate volume states.

Table 9-1**Duplicate Volume States**

Volume State	Duplication Status	Description
Done	Successful	A valid, up-to-date duplicate volume is complete.
	Empty	Duplication is complete but no data was written to the duplicate volume.
	Old	Another backup was appended to the original volume since this duplicate was completed. Therefore, the duplicate is not a complete duplicate of the original.
Active	Empty	Duplication of the backup is in progress. Backup information was appended to the original and was queued for duplication. During this time the duplicate shows up as Active, Empty, or Active, Old. (Use vmdupd -all to verify.)
	Old	Data was appended to the original backup volume but the duplication is in progress.
Failed	Empty	Duplication of this volume failed for some reason, or duplication was intentionally canceled. Check the detail log (/var/adm/epoch/detail) for more information about the failed duplication. Reschedule this duplication.
Suspended		Duplication of the backup volume was suspended.
Resuming		A suspended duplication is restarting or an appended backup is being duplicated and the duplication is starting.
Scheduled		The backup volume is a candidate for duplication.
Imported		The duplicate volume was imported into the EDM and is ready for use.

Importing a Duplicate Volume

If you import a duplicate volume (using **ebimport**), **ebreport media** reports the status as “Not started, empty,” which is erroneous. This status is corrected when the duplicate volume is appended to.

Note: The duplicate volume, when imported, is not substituted for the original volume if the original was modified since the last duplication.

Importing a Duplicate Volume before the Original

If you import a duplicate before the original, a volume catalog entry is also created for the original.

If the original is imported:

1. A dialog asks if you want to delete the existing entry with the same volume ID. You should answer Yes.

The first entry for the original still appears; a second, uncataloged volume also appears in the same drive.

2. Close the Library Unit Manager window.
3. Click on Volumes to reopen it.

Now only the uncataloged volume is in the drive and the first entry for the original volume is gone.

4. Now import the uncataloged volume again.

Rejecting a Mount Request

When you reject a mount request through the Media Request window, the scheduled duplication is set to Failed to prevent the duplication from being recursive. This Failed status appears in the detail log. (See “Log File Rotation and Archival” on page 15-4.) System monitoring also flags this status. (See *EDM System Monitoring Report* for more information.)

You should reschedule a failed duplication by selecting it in the Duplications window, or by using **vmdup -reschedule** at the command line. (See “Viewing Reports on Duplications” on page 9-25.)

Note: It is important that failed duplications be rescheduled, as no subsequent backups to the original are duplicated until rescheduling occurs and duplication is successful.

Expiring a Duplicate Volume

You can expire a duplicate volume at any time before or at the same time you expire an original volume. (The original volume should be taken offsite rather than the duplicate because the expiration period of an original volume can be longer than the duplicate.)

In the Media tab of the Backup Configuration window, click on the Expiration Options button. In the Media Expirations window that appears, set the expiration period in years, months, or days. Then click on Apply.

Note: If you attempt to set the expiration date of a duplicate after that of its original volume, an error message appears. You must specify a new expiration date before you can continue.

To expire a duplicate volume at the command line, use the **ebexpire** command. This command enables you to expire original and duplicate volumes as well as saveset records and backup catalogs. (Refer to the ebexpire man page for more information.)

To expire a duplicate volume, enter the following:

```
# ebexpire -d -expire -purge
```

The -d option specifies that a duplicate is to be expired.

Viewing Duplicate Expiration Dates

You can use either of two commands at the command line to view duplicate expiration dates:

1. Using the command **ebreport history -expire_times** lists all expire times for media, catalogs, duplicate media, and savesets. An example follows:

**** Work Items for Template dlt_dup_2, Primary Trailset ****

**Item "dup:/home/client_2" for client "xyz"

Time	Lvl	ID	Status	Entries	Cat_Exp	Dup_Exp	SS_Exp	Med_Exp	Rcvr
12/04/99 14:13	9	72306582.345F741C	complete	412	12/05/99	12/05/00	03/05/00	03/05/00	r
12/04/99 13:27	9	72306582.345F6924	delta	0	12/05/99	no dup	03/05/00	03/05/00	
12/04/99 11:19	9	72306582.345F4B50	complete	0	12/05/99	01/05/00	03/05/00	03/05/00	
12/04/99 10:59	9	72306582.345F46B8	complete	0	12/05/99	expired	03/05/00	03/05/00	
12/04/99 10:45	9	72306582.345F435B	delta	0	12/05/99	no dup	03/05/00	03/05/00	
12/30/99 12:57	9	72306582.3458CAF6	complete	0	12/31/99	01/05/00	03/30/00	03/05/00	

2. Using the command **ebexpire -check** scans the saveset database and lists the current state of the selected backup resources, including the status of duplications. An example follows:

```
72306582.345F741C: media state is "onsite", expiring on 12/05/99 14:1
72306582.345F741C: saveset state is "active", expiring on 12/05/99 14
72306582.345F6924: catalog state is "delta", expiring on 12/05/99 13:
72306582.345F6924: duplicate state is "exists", expiring on 12/05/99
72306582.345F435B: media state is "onsite", expiring on 12/05/99 10:4
72306582.345F435B: saveset state is "active", expiring on 12/05/99 10
72306582.3458CAF6: catalog state is "delta", expiring on 12/05/99 12:
72306582.3458CAF6: duplicate state is "expired"
```

10 Magnetic Disk Concepts

To keep your system working smoothly, you must monitor disk space and catalogs, and perform some system maintenance. Although EDM Backup software has commands and scripts to control disk space usage, you should also monitor it periodically.

This chapter covers the following topics:

- Expiration of Backups and Catalogs
- Filesystem Cleanup Script
- Magnetic Disk Capacity
- Managing Disk Space

Expiration of Backups and Catalogs

On occasion, you need to expire old backups and backup catalogs. Expiring backups frees up storage media for reuse as well as the disk space that their corresponding backup catalogs use on the server. Expiring additional catalogs of older backups can free up additional needed disk space on the server.

EDM Backup software creates a backup catalog each time it backs up a work item. The backup catalog identifies a backup at the file level by recording the names and attributes of each file in the work item at the time of the backup. It also keeps track of the location of backed up data for each file that was selected for backup. You need the backup catalog when restoring files.

Catalogs are stored online on the server and can grow to be quite large. To maintain sufficient magnetic disk space you must expire the catalogs after a fixed period, perhaps quite earlier than you wish to expire the backups themselves.

You can expire a catalog and still restore data from its corresponding backup if necessary, because unprocessed catalogs are also stored directly on the backup tapes along with the backup data. If you need to access an old backup, you can recreate the catalogs from these raw, unprocessed catalogs on the media by using the **ebimport** command.

For each backup of a work item, the server also creates an online saveset as well as the backup catalog. The *saveset* record contains information about an entire backup, for example, its start time, the media trail that the backup program used to write the backup data, and the expiration periods. The saveset records do not occupy much space.

Choosing Expiration Periods

The Media Expirations pop-up window enables you to set fixed periods for expiration of backup data, catalogs, and the saveset record. You reach this pop-up window through the Media tab of the Backup Configuration window.

You should change the timing for expiring catalogs, backups, and savesets according to these rules:

- Backup period must be greater than twice the rotation period because incremental backups on one tape in a media set (trailset) assume access to the previous incrementals and most recent full backups on previous tapes in that media set.

Note: It is important for all of these expiration periods to be greater than twice the rotation period. The reason for this is that you need a full backup and subsequent delta catalog files to reconstruct your system for disaster recovery. It is especially important if you automate deletion of expired backups in crontab.

- Catalog period must be greater than twice the rotation period, but it can be less than the backup period.
- Saveset period must equal the backup period unless you choose never to expire the backup period. In no case can it be less than twice the rotation period.

You can change the expiration periods at any time. Any changes to the expiration periods take effect the next time **ebbackup** runs, either from an entry in root's crontab file or manually by typing the command **ebbackup** from the command line.

```
# /usr/epoch/EB/bin/ebbackup
```

Running Expiration

After a backup is eligible for expiration, you can delete it from your system, either automatically by running the command **ebexpire -purge** in root's crontab, or manually from the command line. Use the **-purge** option with the **ebexpire** command when you want to delete expired catalogs.

```
# /usr/epoch/EB/bin/ebexpire -purge
```

The **ebexpire -purge** command identifies backup data, catalogs, and saveset records that are eligible for expiration because they exceeded their duration period as set in the backup configuration. (Refer to the **ebexpire** man page for more information about this command.)

Filesystem Cleanup Script

Try to keep the number of old and unneeded files to a minimum on your system to conserve space for the backup catalogs. The cleanup script **epcleanup** simplifies the cleanup of your filesystem by removing unneeded or old files from tmp, crash, and adm subdirectories.

The system is automatically configured to execute the **epcleanup** script from **cron** with the default settings. The line in root's crontab file that specifies the **epcleanup** script is:

```
30 8 * * * /usr/epoch/lib/epcleanup > /dev/null  
2>&1
```

If you want to override any of the defaults, specify the option name and the new value in the crontab file. For example, if you want to change the number of days to expire unmodified log files from 14 days (the default) to 10 days you enter:

```
30 8 * * * /usr/epoch/lib/epcleanup -log 10 >  
/dev/null 2>&1
```

Magnetic Disk Capacity

You must know your system's catalog capacity to be able to configure your site for optimal performance.

EDM Backup software consumes magnetic disk space based on the total number of files to back up. In addition, disk space consumption is affected by the rotation period (the number of days between level 0 backups), the expiration period for catalogs, the percentage of files that change on a daily basis, and the average catalog size per file.

The calculations in this section assume that:

- the system is used as a backup server only.
- five percent of the files are modified each day; this affects incremental backups (see "Calculating Actual Daily File Changes" on page 10-7).
- catalog size averages 200 bytes per file.
- catalog requirements for database backup are small.

Rotation and Keep Catalog Periods

Keep in mind that the *minimum* keep catalog period must be at least twice that of the rotation period to ensure that the catalog is relieved of its dependencies. The system defaults are a rotation period of 14 days and a keep catalog period of one month.

Table 10-1

Minimum Keep Catalog Period (Days)

Rotation Period		
7 Days	14 Days	28 Days
14	28	56

Table 10-2 shows the *maximum* number of days that you can use for the keep catalog period for a 25.2 GB catalog subsystems as you vary the number of files that you back up and the rotation period.

Pick the table that matches the catalog disk size that you have. Then select the number of files to back up. You can then refine the suitable rotation period and keep catalog policy.

As long as you are well within the maximum limit of files, you are free to adjust rotation and keep catalog periods up and down. As you approach the upper limit of files, you run into some constraints, as noted in Table 10-2.

Table 10-2

25.2 GB Catalog Subsystem: Max. Keep Catalog Period (Days)

Number of Files (Millions)	Rotation Period		
	7 Days	14 Days	28 Days
1	470	745	1045
2	235	370	520
3	155	245	350
4	115	185	260
5	95	150	210
6	80	125	175
7	65	105	150
8	60	90	130
9	50	80	115

Table 10-2

25.2 GB Catalog Subsystem: Max. Keep Catalog Period (Days)

	Rotation Period		
10	45	75	105
12	40	60	85
14	35	55	75
16	30	50	70
19	25	40	56
23	20	30	< 56 ¹
31	15	< 28 ¹	< 56 ¹

1. Not allowable because the maximum is less than the minimum of 2x the rotation period.

Calculating Actual Daily File Changes

Table 10-1 and Table 10-2 above assume that five percent of the files are modified each day. By looking at a backup history report, you can calculate the actual number of files that change each day at your site.

This affects the incremental backups. If the actual percentage is less, additional files can be backed up. If the actual percentage is more, fewer files can be backed up.

You must wait until EDM Backup has been running for one rotation period before you can generate enough information to determine the number of daily file changes.

The catalogs for incremental backups are consolidated to reduce storage space on the server. All but the most recent incremental backup catalog are turned into *deltas*, which list only backup files that differ from those that are listed in subsequent catalogs.

Note: If you custom schedule an explicit incremental backup (level 1-8), by default, the catalogs for these backups are *not* consolidated. You can change the *backup catalog delta level* field in the Backup Configuration window from 9 to the lowest integer level for which you want consolidated catalogs.

You must review the number of entries in a *delta catalog* to determine your daily changed file rate. A delta catalog contains only the information that differs from the subsequent catalog files. The number of entries in a delta catalog represent the number of files that were changed during the time between the backup that delta represents and the subsequent backup. Thus, the size of a delta catalog represents a measure of how many files were changed during a backup.

Run a backup report (**ebreport backup**) to display the number of files or directories (entries) in each backup catalog.

Figure 10-1 illustrates a report that was produced with the **ebreport backup** command, and specified with the **-since** option and a date. The report shows the type of backup catalog that was created on this date, and the number of file entries.

Figure 10-1**ebreport backup -since Report**

EDM Backup Backup Report for server adam at Oct 24 13:44:05 1998
 Report options: -since 9/15/98

Template name, Primary/Alternate: Trailset name
 =====

default, Primary: primary

Work item name Catalog	Level	Start time	Time used	BackupFiles\bad	Size

adam:/	9	9/23/98 19:33:03	0:12:22	Completed	2695\0214
MB Unsorted					
adam:/	9	9/22/98 19:42:24	0:13:42	Completed	926\0115.
MB Complete					
adam:/	9	9/21/98 18:14:37	0:11:24	Completed	234\00.01
Delta					
adam:/	0	9/21/98 15:44:04	0:43:48	Completed	26720\0

To determine the number of files that are backed up each day, run a backup history report for a complete rotation period, not including the last backup run. For example:

emc# **ebreport history -since 9/6/98 -until 9/19/98**

This command produces a report that covers a 14-day rotation period (refer to the **ebreport** man pages for more information). The command output provides a line for each backup of each work item, which shows the catalog type and the number of entries in the backup catalog.

For example, you can view a report that contains lines that are similar to those in Figure 10-2.

Figure 10-2**ebreport history -since -until Report**

Backup History Report for server atlas1 on Nov 20 16:06:15 1997

**** Work Items for Template cad-all, Primary Trailset ****

**Item "cad5-all"

TimeLvlIDStatusEntriesExpire

1/ 7/97 20:010C005531B.296A4F13complete317661/ 6/98

1/ 6/97 20:019C005531B.296A32E9delta39431/ 5/98

Backup Status (delta or complete)

Managing Disk Space

To keep your EDM system working smoothly, you must monitor your magnetic disk space. This section describes several ways to do this.

Distributing Catalogs

After you run EDM Backup software for awhile, you must split the backup catalogs proportionately among the available space on each disk. EDM Backup software creates catalogs for each backup to track the file data in the backup. Initially, EDM Backup software places all backup catalogs in a single partition under the directory (or symlink) /usr/epoch/EB/catalogs.

The installation procedure creates the directory /usr/epoch/EB/catalogs. In the catalogs directory each work item has one subdirectory – named for that work item. EDM Backup places all catalogs produced for that work item in that subdirectory, even if the work item is backed up through multiple templates and trailsets.

Note: You must keep the catalogs on disks that are local to the backup server, not on an NFS-mounted filesystem.

To split the backup catalogs among the disk partitions, use the following procedure:

1. Run **ebbackup** until every work item had at least one successful backup through each of the configured schedule templates and trailsets (media sets).

Use the **ebreport history** command to determine that all work items had a successful backup. For information on using this command, see the **ebreport** man page.

2. Divide the work items into one group per disk partition so that the total number of catalog entries in each group is in proportion to the size of the partitions. You can determine catalog entries per group from the **ebreport history** report output.

To determine the number of catalog entries in a work item, run the **ebreport history** command. For each work item, add the number of entries from the most recent complete catalog for every template and trailset. This sum is the size of the backup catalogs that EDM Backup places in the work item's directory during a rotation period.

For example, consider a site that has 50 work items of the same size (CAD1 through CAD50) and three available disk partitions for backup catalogs. Two of the disk partitions have 900 MB of free disk space and the third disk partition has 400 MB. To divide the total catalog entries proportionately to the ratio of available disk space, the site places 21 work items on each of the 900 MB partitions, and eight work items on the 400 MB partition. Figure 10-3 below illustrates this distribution.

Figure 10-3

Work Item Distribution



3. Move the individual work item entries for each group of work items onto the target disk partition.

Note: Do not move a work item directory while backups are running or while catalogs are being processed.

For example, assume that `/usr/epoch/EB/catalogs` is a symbolic link to `/home/EB/catalogs` and that all catalogs were initially on the `/home` partition in that directory.

To move the work items `cad10` through `cad20` from `/home/EB/catalogs` on the disk partition `/home` to the disk partition `/data1`, move the individual subdirectories for each of these work items from `/home/EB/catalogs` to `/data1/EB/catalogs`.

For example, using the Bourne shell, type:

```
edm# mkdir /data1/EB
edm# mkdir /data1/EB/catalogs
edm# cd /usr/epoch/EB/catalogs
edm# sh
# for wi in cad1? cad20; do
> tar cf - $wi | (cd /data1/EB/catalogs; tar xf -)
> rm -fr $wi
> ln -s /data1/EB/catalogs/$wi .
> exit
```

4. Monitor the storage usage of the partitions. If one partition fills up, move the work item subdirectories to the other partitions to keep the proportions balanced.

Do not move a work item directory while EDM Backup is running or while catalogs are being processed. Always make sure that `/usr/epoch/EB/catalogs` has a symbolic link to the catalogs in the work item directory. Also, you must keep catalogs on disks that are local to the server. For example, catalogs must be on a local filesystem and not on an NFS mounted filesystem.

Reclaiming Magnetic Disk Space

If you do not have HSM, when your magnetic disks fill to capacity, your backups cannot complete successfully. You know that you exceeded magnetic disk capacity on your system when:

- you see numerous messages similar to:
`edm: /home: file system full`
- you see numerous messages that backups have failed.
- the **df** command shows that one or more of the local filesystems used over 100 percent of its magnetic disk space.

Manually Expire Unneeded Catalogs

To create space on your disks and to get backups running again, use the following procedure:

1. Run the **ebcatclean** command to expire unneeded catalog related files.
 - If running this command gives you enough magnetic disk space to operate, you can stop at this point.
 - If running this command does not provide enough disk space to operate, continue with the following steps to provide additional space.
2. Run **ebreport history** to see the state of all backup catalogs. You want to ensure a complete catalog to support the delta catalogs you save.

3. Choose a date for expiring backup catalogs. Do not expire up to within two times the rotation period.

For example, if the rotation period for the specified schedule template is 2 weeks, make sure that the **-until** date (in **ebexpire in the next step**) is at least 4 weeks plus one day ago.

4. Run **ebexpire** with the **-c** switch (to expire the backup catalogs without expiring their backups), and the options **-purge**, **-template**, **-since**, and **-until** for the specified schedule template during the specified dates.

CAUTION: If you do not specify the -c option, you will also expire the backup data and the saveset record, which means that the backup is completely unrecoverable.

If you expire the backup catalogs without expiring their backup, you will still be able to restore your data. Unprocessed catalogs are stored on the backup tapes. You can bring the catalogs online with **ebimport** and then restore the data.

Other Options

The default configuration places a relational database on the /usr filesystem. If that filesystem fills up, and you cannot identify any files to delete or relocate, you can try one of the following:

- Rebooting to free any space that may have been used by dead processes to hold open deleted files
- Expiring backup media and relocating the database to a filesystem other than /usr
- Increasing the size of the /usr filesystem

If necessary, contact customer service for assistance.

Changing the Automatic Cleanup Script Defaults

Try to keep the number of old and unneeded files to a minimum on your system to conserve space for the backup catalogs. The cleanup script **epcleanup** simplifies the cleanup of your filesystem by removing unneeded or old files from /usr/epoch/tmp, /usr/epoch/adm, and /usr/epoch/etc.

The system is automatically configured to execute the **epcleanup** script from **cron** with the default settings. The line in root's crontab file that specifies the **epcleanup** script is:

```
30 8 * * * /usr/epoch/lib/epcleanup > /dev/null 2>&1
```

If you want to override any of the defaults, specify the option name and the new value in the crontab file. For example, if you want to change the number of days to expire unmodified log files from 14 days (the default) to 10 days you enter:

```
30 8 * * * /usr/epoch/lib/epcleanup -log 10 > /dev/null 2>&1
```

Part II Hierarchical Storage Management (HSM)

11 Basic HSM Concepts

EDM Backup with HSM Option and EDM Migration client software extend filesystem space by migrating file data on magnetic disks out to optical disks, magnetic tapes, or even other magnetic disks, which creates a much larger *virtual* filesystem. All files, even those that are staged out, appear to the user to be resident on the local magnetic disk.

EDM Backup with HSM Option software provides HSM for the local server. EDM Migration client software extends HSM support to network clients. To enable network migration, you must configure HSM on both the EDM and the network clients.

This chapter introduces some basic migration terms and concepts that provide you with background information for performing tasks.

These concepts include:

- When Files Stage In and Out
- Filesystem Configuration and Maintenance
- File Control Properties
- Compaction of Staging Media
- Compacting Baseline Media

- Migration Reports
- Baseline Backup
- Restaging Data
- Backup Completeness

When Files Stage In and Out

There is a limitation of a maximum of 2GB minus 1KB for the size of files to be staged out or staged in.

A file is staged out:

- as part of nightly system maintenance. This is called *periodic* stage out because the staging occurs on a schedule (see Figure 11-1). You set up nightly staging runs to reduce disk utilization to a predetermined level, called the low watermark (LWM). You can set up periodic staging runs through root's crontab file.
- during daily system operation when magnetic disk space usage reaches a predetermined level called the high watermark (HWM), or when the filesystem runs out of disk space. This is called *event-driven* or *demand* stage out because it is triggered by a growth in disk usage.
- in response to a user's request. Users can explicitly stage out files with the **emstage** command and stage in files with the **embsi** command.

HSM stages a file back in when:

- the staged-out file is read, or
- the staged-out file is modified

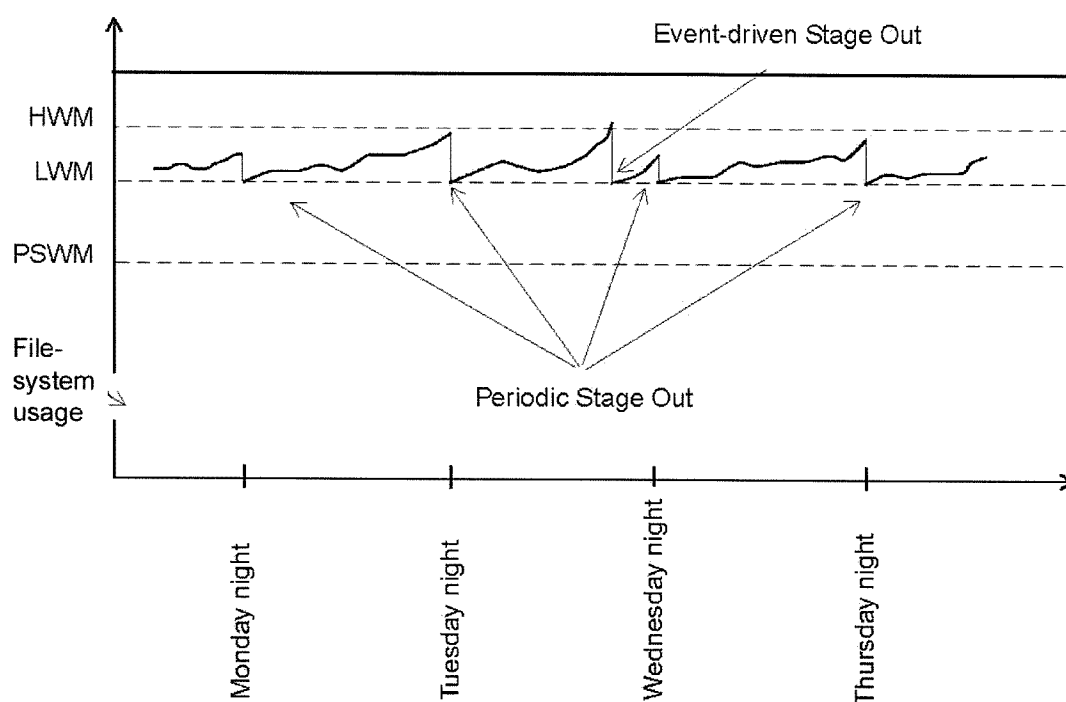
When a user reads a staged-out file, migration locates the file and stages it in. At this point, the file is considered *prestaged* – that is, the file resides on magnetic disk *and* on the staging media. Migration may later free the local magnetic storage for this file, without having to repeat the stage out process.

When a user modifies a staged-out file, migration stages in the file and then deletes the staged-out version. If migration later needs to reclaim magnetic storage, it must stage out this file again.

Figure 11-1 shows event-driven and periodic staging over a four-day period.

Figure 11-1

Staging Out File



Filesystem Configuration and Maintenance

Configuring HSM is a matter of determining how you can best tune your filesystems so that the most active files are readily accessible. This section discusses several basic concepts that are pertinent to configuration, including:

- staging templates and staging trails

- bitfiles and client stores
- watermarks
- periodic staging and filesystem delay

Staging Templates and Staging Trails

A staging template defines default values for the filesystems that stage to a particular *trail* of optical disks or tapes, or in the case of network migration, a particular *client store* on the server. These default values include:

- template name
- trail type (EO, DLT, network, etc.)
- volume availability (restricted or unrestricted), which specifies how volumes that belong to this template can be reused after they are compacted. *Restricted* volumes can be reallocated only to the same template.
- self-describing media enable/disable

A staging template can also define configuration values such as watermarks and a delay factor. Staging templates exist for filesystems on the migration server and for filesystems on network clients.

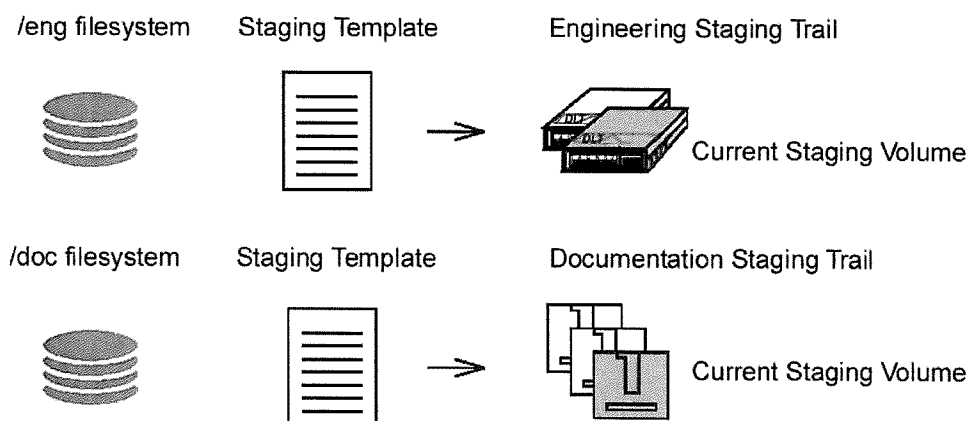
When files on the migration server are staged out, they are written to *a staging trail*. Initially, a staging trail consists of a single side of an optical disk or a magnetic tape. Over time, a staging trail can grow to include several optical disks or magnetic tapes. The piece of media that is currently being staged is called the staging trail's *current staging volume* (see Figure 11-2). Staging trails usually share the same name as the staging template.

When the current volume is full, the Media Requests window tells you to add a new, blank volume to the staging trail. Eventually, this volume fills up as well, and the process is repeated. Over time, this process builds up a trail of staged-out

files for all of the filesystems that are assigned to a template. (Refer to the section on Labeling Volumes for information on allocating a volume to the staging trail.)

Figure 11-2

Staging Templates and Associated Filesystems



Deciding How Many Staging Templates to Create

Depending on your site's usage patterns and needs, you can assign all filesystems to a single staging template or assign certain filesystems to their own staging templates. The simplest way to set up staging is to group your filesystems into the fewest number of staging templates as possible. This has several advantages:

- It directs all of your files to a limited number of staging volumes. This reduces the number of staging volumes that need to be moved in and out of library units. When a user requests a staged-out file, there is a greater likelihood that the file resides on the volume that is already in the drive.
- It reduces competition for staging devices. If multiple filesystems begin to stage out at the same time, they can compete with each other for access to staging devices,

leading to a phenomenon called *thrashing*, where the system must repeatedly swap media in and out of drives. Sharing a template prevents this from happening.

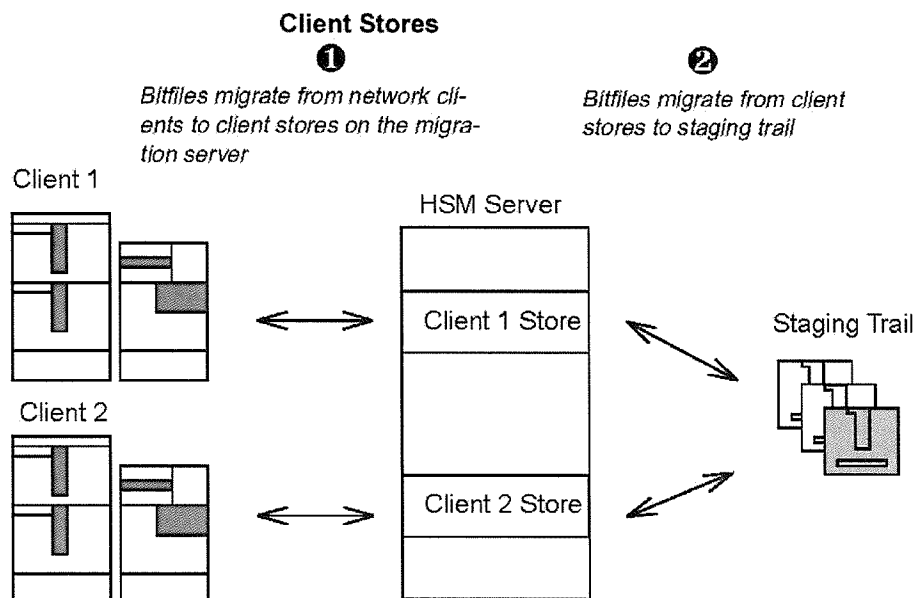
However, there are cases when it is advantageous, even necessary, to create multiple staging templates:

- When you stage files of widely differing sizes. You can reduce the number of mount faults for the smaller files if you keep very large files on a separate staging trail.
- When your site has different access patterns (for example, archival vs. working data). Archival data should be staged separately from working data.
- When you want filesystems that are separate on magnetic disks to also be separate on staging media.
- When you charge groups for the costs of storage media and/or storage space. To keep track of costs, you can create staging templates for each business group and charge them separately.
- When your site expects to expand to multiple servers and move some of the data to a new server. In this case, a separate staging template for each anticipated server simplifies moving the data.

Bitfiles and Client Stores

EDM Migration client software is responsible for the movement of files between network migration clients and the migration server. EDM Migration automatically maintains the relationship between local storage on the client system and server storage, and keeps track of all file data independent of its location.

Figure 11-3



What EDM Migration client software actually stages is a *bitfile* – an uninterpreted bit array. A single bitfile holds the contents of a single client file. Bitfiles are staged to the migration server where they are kept in permanent administrative groupings called *client stores*. Each client system owns one or more client stores on the server. Other clients may be permitted to read these bitfiles, but only the owner client can create or delete the bitfiles.

Every client store is associated with a *store ID*, which uniquely identifies it on the network.

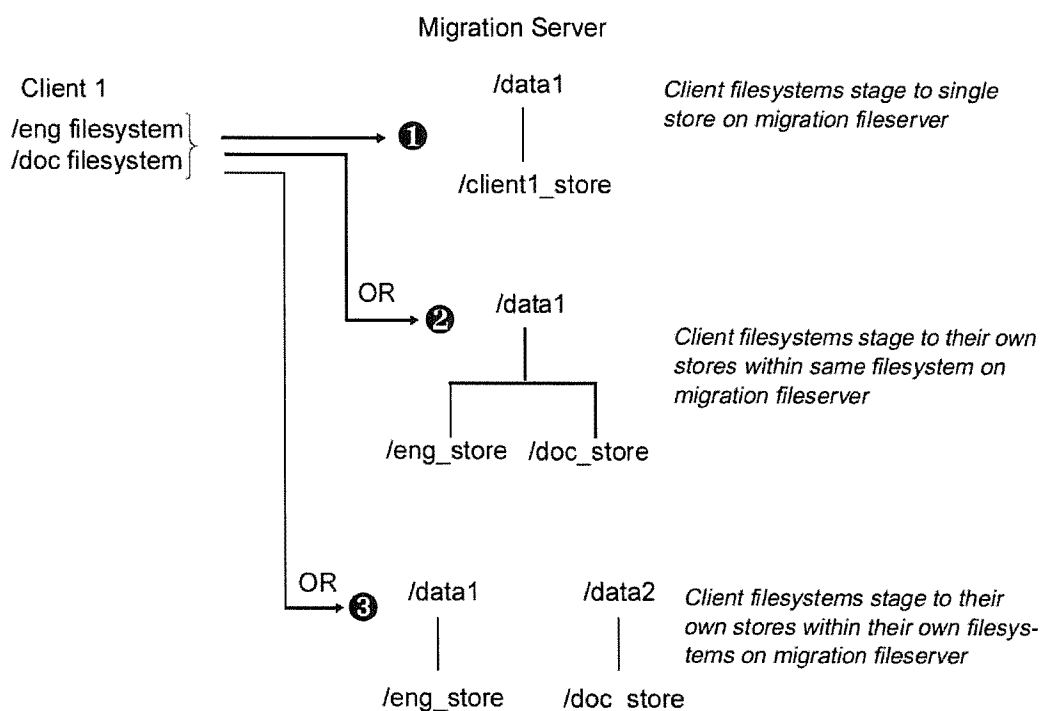
Deciding How Many Client Stores to Create

A client store can reside within any migration-enabled filesystem on the migration server. Client stores can all be grouped into a single filesystem on the migration server or spread out among several filesystems. If a network client contains several stageable filesystems, you can stage all of the filesystems to a single store on the migration server (Figure 11-4, example 1); you can stage each filesystem to its

own store within the same filesystem on the migration server (Figure 11-4, example 2); or you can stage each filesystem to its own store within its own filesystem on the migration server (Figure 11-4, example 3).

Figure 11-4

Client Store Configurations



As a general rule, the simplest configuration appears in Figure 11-4, example 1, where all of a client's filesystems stage to a single store on the server. If you have three network clients, for example, you would only need three client stores, one for each client. All of the stores would reside within the same filesystem on the migration server.

However, there are cases where it would be advantageous, or even necessary, to distribute the client stores across filesystems. In fact, most of the reasons for creating multiple staging templates (see “Deciding How Many Staging Templates to Create” on page 11-5) are also true for client stores. In addition:

- If a client filesystem consists of a large number of small files, it should stage to a store in its own filesystem. Otherwise, it could cause a filesystem on the server to grow beyond the limit of one million files.
- If you expect there may be a future need to move a client filesystem to another client, you should have that filesystem stage to its own store. This simplifies the move because two clients cannot stage to the same store.

Preventing Redundant Backup of EDM Migration Client Data

If migration client stores are being backed up from the server, there is no need to back up the corresponding data on the client itself. Use the Backup/HSM tag to prevent migration client data from being backed up redundantly.

The Backup/HSM tag is used to update the backupdates file (/usr/epoch/etc/backupdates). When a backup begins for a server work item that has a Backup/HSM tag, a time stamp is entered in the backupdates file. Before a migration client is backed up, the backupdates file is checked. If a file exists on the client that is newer than the client store’s work item date listed in backupdates, the file is backed up.

From the EDM configuration interface, you set the tag’s value in two places: Backup configuration and HSM configuration.

When you define the work item that backs up the client store filesystem on the EDM (Backup Configuration window, Work Item tab, Work Item options, HSM Options), enter an identifier in the Backup/HSM Tag field. EMC recommends that you use the work item name for the tag. The name must be unique across all work items that back up the EDM’s files.

In the HSM configuration window, Client tab, select the tag that you entered in backup configuration for the work item that backs up the store.

If you set the tag from the command line (the `emsmks` command), note that the tag must match exactly the backup/HSM tag name specified in the definition of the work item.

Full Filesystems

When a magnetic disk becomes 100% full, the system logs a “filesystem full” message via **syslogd**. By default the message is displayed on the console and recorded in the system log file. Being full should be a transient condition for stageable filesystems. Because files are staged out when usage reaches the high watermark, stageable filesystems should rarely fill up.

Watermarks

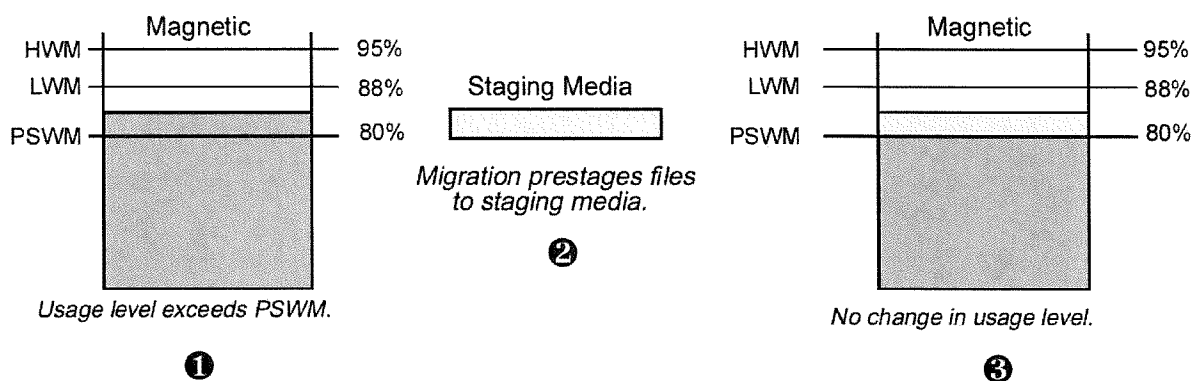
Migration keeps disk space utilization between user-configurable usage levels called *watermarks*. Watermarks are thresholds that trigger a migration response. Watermarks are expressed as percentages of a filesystem’s total disk space, minus the space that some systems withhold from ordinary users. The three watermarks are:

- High Watermark (HWM)
- Low Watermark (LWM)
- Prestage Watermark (PSWM)

To understand how watermarks regulate file migration consider the following example. Assume you have recently installed your migration server and your filesystems still have significant amounts of free space. When users add enough files to cause the usage level to exceed the PSWM, the nightly periodic staging run (set up automatically) will stage files out to the staging media, without freeing any space from the magnetic disk. This is called *prestaging* (see Figure 11-5). Because the

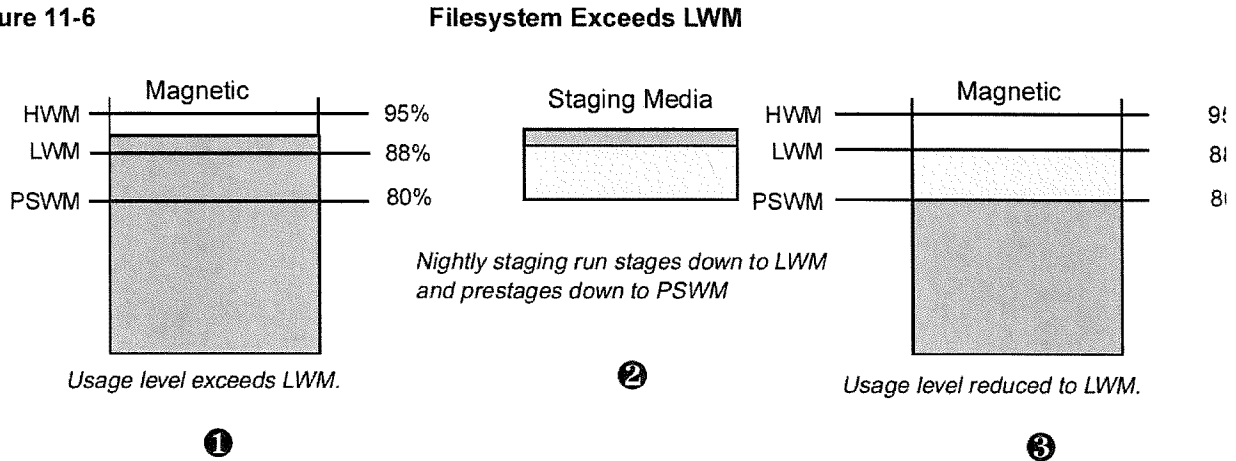
prestaged files still reside on magnetic disk, users can access them quickly. If filesystem usage rises dramatically, migration can simply remove the magnetic images of these prestaged files.

Figure 11-5

Filesystem Exceeds PSWM

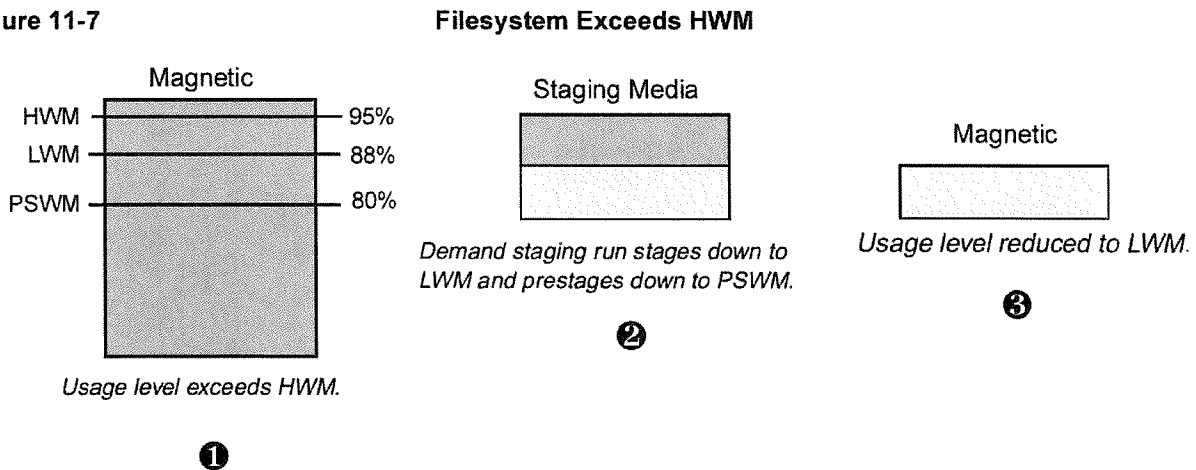
When your users add enough files to cause the usage level to exceed the LWM (88% full), the nightly staging run stages out enough files to bring usage down to the LWM (see Figure 11-6). To do this, migration actually moves files from magnetic disk to secondary storage. In addition, migration prestages files down to the PSWM.

Figure 11-6



When your users add enough files to cause the usage level to exceed the HWM (95% full), migration triggers a demand staging run, that is, migration immediately stages out enough files to bring usage down to the LWM (see Figure 11-7). Again, migration actually moves files from magnetic disk to secondary storage and prestages files down to the PSWM.

Figure 11-7



Generally speaking, once you have enough files to fill your magnetic disks, your goal is to keep filesystems in the *green zone*, that is, between the low and high watermarks. A key decision, then, is how large to make the green zone. The optimal size of the green zone depends on your filesystem's usage patterns.

Table 11-1 shows the watermarks for a 1000 MB filesystem. On this filesystem, stage-outs begin when disk utilization reaches 95% capacity, or 950 MB; stage-outs stop when disk utilization falls to 88% capacity, or 880 MB; and migration prestages another 80 MB until disk utilization falls to 80% capacity, or 800MB.

Table 11-1

Watermarks for a 1000 MB Filesystem

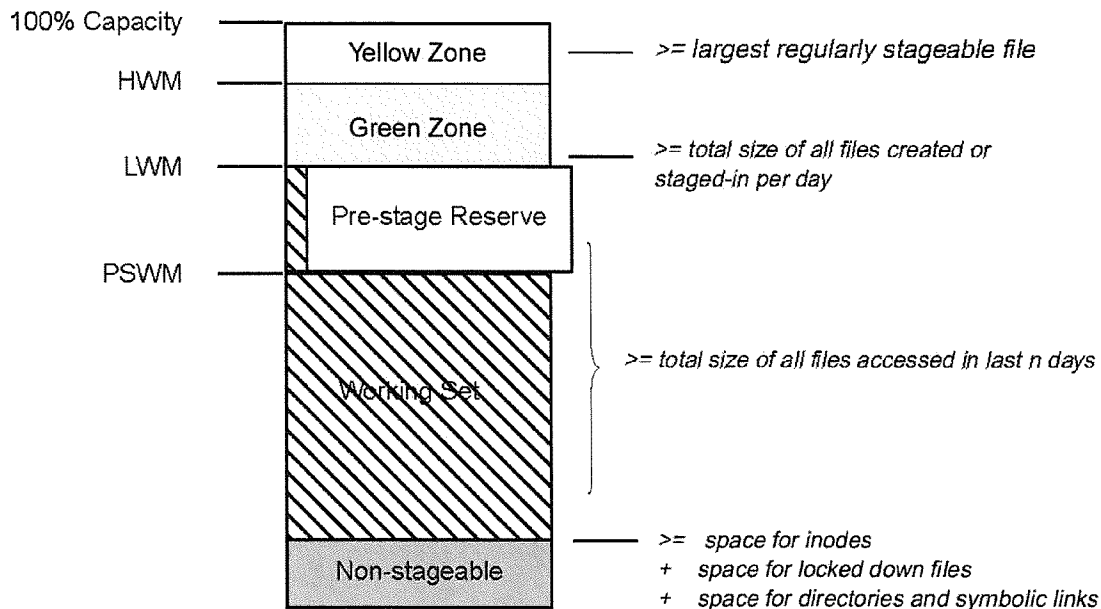
Watermark	% Capacity	# of MB
HWM	95%	950 MB
LWM	88%	880 MB
PSWM	80%	800 MB

Disk Utilization Zones

The watermarks divide the filesystem into disk utilization zones (see Figure 11-8). The space between 100% capacity and the HWM is the yellow zone, the space between the HWM and the LWM is the green zone, the space between the LWM and the PSWM is the prestage reserve, and the space between the LWM and the non-stageable data is the working set.

Figure 11-8

Utilization Zones



The *yellow zone* is reserved for processes to use while migration brings filesystem usage back down to the LWM. It represents the area between 100% capacity and the HWM. Note that if you make your yellow zone slightly larger than the largest regularly stageable file in your filesystem, the system can use this space to stage in or create most any file immediately, while migration then frees additional space by staging out other files, usually from the prestage reserve.

The *green zone* represents the area between the HWM and the LWM, and is the normal zone of operation. The green zone should be large enough to hold the average number of new disk blocks added in a day, including both new files and previously inactive (staged-out) files that are likely to be accessed (staged-in). It should be large enough to make event-driven staging infrequent. Making the green zone larger or smaller is the most common change to the default configuration.

The *prestage reserve* is used for files that have been staged out, but also remain on the system's magnetic space. This magnetic space can be released quickly if disk utilization crosses the HWM. To allow filesystem usage to return to the LWM during a demand-staging event, the prestage reserve is typically the same size, or slightly larger, than the green zone.

The *working set* represents the files that are accessed in a given period of time. For general applications the working set should be in the 7-30 day range. The magnetic disk space holding the working set is actually a combination of the prestaged reserve space (since these files, although staged, are still magnetic resident) and the amount of space available to stageable files below the PSWM. This area needs to be large enough to contain all files accessed in the last n days, where n is the number of days worth of recently accessed files that you want to fit within the working set.

Non-stageable files and other disk structures also consume space; these include space for all directories and symbolic links or for any other files that cannot be staged (for example, swap files).

Sample Watermarks

Consider using one of the sets of watermarks listed in Table 11-2 for your configuration.

The *Archive* settings are designed for filesystems whose files are written once and rarely, if ever, read. The filesystem's data is typically staged-out and rarely, if ever, staged back in. This is the case if large amounts of data are gathered every day and quickly "archived" off of the magnetic disk.

The *Cached* settings are designed for filesystems in which reads outnumber writes, and a relatively predictable set of files are read. This setting takes advantage of migration's ability to keep the most recently-accessed files on magnetic disk, thus ensuring optimal performance.

The *Random* settings are also intended for situations where reads outnumber writes, but where the access pattern is random and least-recently-used caching is ineffective. This would be the case, for example, in a government records office, where several files must be read in from staging media in order to analyze a new file. When the analysis is completed, there is no need to keep the files on magnetic disk, because the files are not accessed again for an undetermined period of time. You can select this setting for filesystems that match this random data access pattern.

You can use these settings as a guide for configuring your filesystems.

Table 11-2 shows the sample watermark settings.

Table 11-2

Sample Watermark Settings

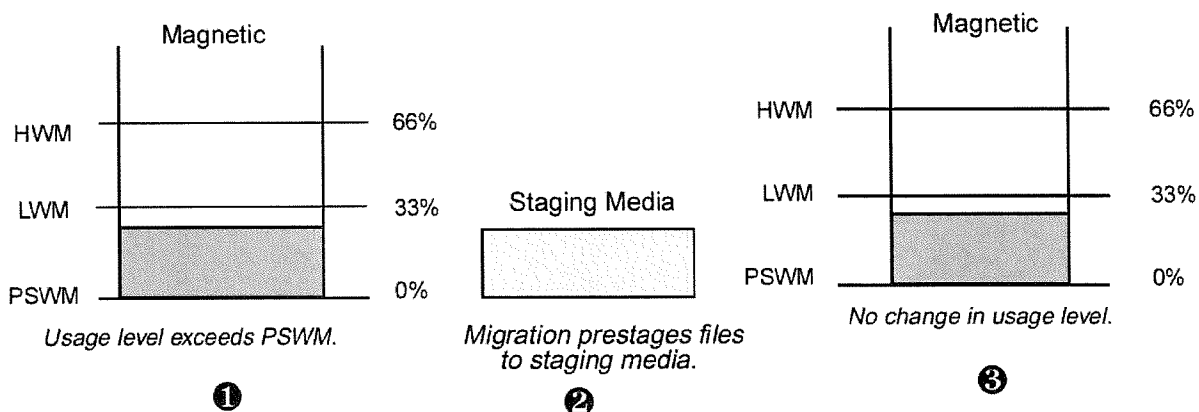
Watermark	HWM	LWM	PSWM
Cached	95%	88%	80%
Archive	66%	33%	0%
Random	68%	34%	17%

The watermarks for Cached filesystems are calculated to maximize the magnetic cache and minimize stale space on staging media.

For archive filesystems, the sample watermarks divide magnetic disk space into three equal zones by setting the high, low, and pre-stage watermarks at 66%, 33%, and 0%, respectively. Any stageable file that is moved onto the disk is prestaged during the next periodic or demand staging run.

Figure 11-9

PSWM Set at 0%



If an archive filesystem contains some temporary files that remains on magnetic disk, you should increase the prestage watermark to take this into account. For example, if a 100 MB archive filesystem is expected to have 20 MB of temporary files, you can set the PSWM at 20% and divide the remaining 80% into equal zones of 26.6%. Thus, the LWM is set at about 47% and the HWM at 74%.

Table 11-3

Watermarks for Archive Filesystems

Watermark	HSM	LWM	PSWM
Default	66%	33	0%
With 20% Temp Files	74%	47%	20%

For filesystems where the random access pattern predominates, the tuning is somewhat similar. The major difference is that when the magnetic disk gets full, it is filled with prestaged files that have been just staged in, whereas archive filesystems are filled with new files. The PSWM is set at 17%. If you expect to fill more than 17% of the filesystem with temporary files, raise

the PSWM and create equal-sized yellow and green zones and a prestage zone of about half the size of the green zone. For example, if a 100 MB Random filesystem is expected to have 25 MB of temporary files, you can set the PSWM at 25%, the LWM at 40%, and the HWM at 70%.

Table 11-4

Watermarks for Random Filesystems

Watermark	HWM	LWM	PSWM
Default	68%	34%	17%
With 25% Temp Files	70%	40%	25%

Configuration Issues

Some issues that you need to consider when setting up HSM are the number of files in a filesystem, the type of media you plan to use, and whether or not to enable self-describing media.

Filesystem Limits

Although EMC's HSM products provide virtually unlimited disk space, all filesystems, even stageable ones, are limited by the number of files, directories, symbolic links, and devices they

Table 11-5 Reasons to Set Filesystem Limits

If you expect a filesystem to consist mostly of many smaller files, or to contain many links, you may find that the filesystem is in danger of running out of inodes. To prevent this from happening, you can decrease the inode density, that is, the number of bytes per inode, at the time of filesystem creation.

HSM supports staging to EO, WORM and DLT. In terms of durability and performance, optical disks are the best choice. Although tapes are the most cost-effective media, they are not recommended for applications that require frequent staging-in of data.

- In archival environments (where there are very infrequent stage-ins)

- For restaging infrequently-used data to a secondary staging device
- For baseline backups (which use migration technology)

The limitations of tape are:

- Tapes are less durable than optical disks. Whereas DLT tapes are good for tens of thousands to hundreds of thousands of passes, optical disks have no pass count limit.
- DLT tapes have an archive life of around 30 years, but optical disks have an archive life of 25 to 100 years, depending on the manufacturer.
- Tapes are slower. Whereas stage-ins from optical library units usually take less than 20 seconds, stage-ins from DLT tape can take several minutes on a relatively idle library unit, and much longer on a system with high stage-in activity.

Important variable settings for stage-to-tape are described in “Tuning for Staging to Tape” on page 13-3.

Refer to Table 11-6 for a description of the tradeoffs in using stage-to-tape.

Table 11-6 **Stage to Tape with Tape Library Units**

	Primary staging device	Primary staging device	Secondary staging device	Baseline backup	Migrate backup catalogs to tape¹
	1 drive	2 or more drives	1 or more drives	1 or more drives	1 or more drives
EMC Policy	Disqualified	Conditional	Conditional	Supported	Conditional
Performance	Only capable of servicing ~15 stage-in requests per hour.	Two drives capable of servicing ~30 stage-in requests per hour. Four drives capable of servicing ~60 stage-in requests per hour.	Good, since optical, not tape, will service most user stage-in requests.	Good	Insignificant, since only catalogs are staged.
Media Wear	Significant problem unless used in real archive applications.	Significant problem unless used in archive applications.	Not an issue if data moved to the TLU is accessed infrequently.	Insignificant. Should create low tape access schedule.	Insignificant, since access rate is low.
Deadlock Potential	Significant	Significant	No deadlock cases	No deadlock cases	Reduced, since staging of catalogs is low frequency activity. Can be minimized by careful scheduling of backup and HSM applications.

1. To eliminate thrashing, at least two drives are required, one drive to read in staged data and one drive to write backups to tape.

Self-Describing Media

With self-describing media enabled on the migration server, migration stores the full pathnames of migrated files on the staging media. This allows the media to be moved from a migration server to another server. With self-describing media enabled, however, migration will require more time to stage files.

You can enable or disable self-describing media with the **emstconf** command.

Periodic Staging and Filesystem Delay

As part of your nightly maintenance, you should schedule periodic staging runs of your filesystems to bring disk utilization down to the LWM. Periodic staging runs are set up through root's crontab file.

If you're staging out files from more than one filesystem, you should stagger migration to minimize loads on your system. The actual time that staging begins for each filesystem depends on the filesystem's *delay* parameter, which you can set with the **emfsconf** command. The filesystem delay parameter specifies the number of minutes to wait after the nightly staging run is scheduled to start before beginning stage-outs for a given filesystem.

In setting the delay, you should also consider backup schedules. Generally, you should schedule backups to run after periodic stage-outs.

File Control Properties

File control properties influence, and in some cases determine, the selection of files to stage out. You can list file control properties and file sizes with **emls -l**. You can change file control properties with **emchmod**. The file control properties are listed in Table 11-7.

Table 11-7

File Control Properties

Property	Description
Locked	Locks the file onto magnetic disk; never stage out the file.
Convenient Stage Out	Stages out the file at the next convenient time, probably during the next periodic run.
Keep	Used in conjunction with convenient stage out to cause files to be prestaged rather than fully staged out.
Residence Priority	Prioritizes the importance of keeping the file on disk. All other things being equal, files with lowest priority are staged first. Priorities are expressed as integers. The highest priority is 1; the lowest priority is 63.

Directories have two sets of properties: one set applies to the directory itself (Directories are not staged out, so setting a directory's own properties has no effect), and the other is the inheritable set. When files are created, their own properties are inherited from the parent directory's inheritable set. When subdirectories are created, both the parent directory's own set and its inheritable set are inherited. Thus, file control properties are passed down through directory trees.

The residence priority remains set when a file is staged out and back in. The convenient and keep properties do not.

Although you should use all properties with care, be especially careful with the lock property. Choosing to lock some files on magnetic disk to increase the performance of one application could result in system-wide performance degradation. Locking too many files can prevent migration from working at all. Before locking files onto magnetic disk, try small changes to the residence priority. Monitor the system carefully to determine the effect both on the application and on the system as a whole. Be careful about setting the inheritable lock property on a directory. All files and directories that are created below that point inherit the lock property.

Listing and Changing File Control Properties

You can list file control properties and file sizes with **emls -l** (see Table 11-8 for flag names). The following example lists file control properties and file sizes for all of the files in the archive directory:

```
edm% emls -l archive
```

Mag KB	Stg KB	I-flags	Flags	Staging media	Volume barcodes	Filename
1024	0	----	0	1---	60	filexyz
24	898	----	0	----	0	fileabc
1	0	--CK	60	----	0	dirabc

In this example:

- The file “filexyz” uses 1024 KBs of magnetic disk space and is locked on the magnetic disk.
- The file “fileabc” was staged out to volume #002-a on staging trail Archive, where it uses 898 KBs of disk space. A fencepost of 24 KB remains on magnetic disk. (A *fencepost* is the portion of a staged-out file that remains on magnetic disk.)
- Any files created in the “dirabc” directory are prestaged at migration’s earliest convenience.

Directories have a set of inheritable properties, which are displayed in uppercase letters in the *I-flags* column. Both regular files and directories have a set of staging control properties that apply to the file or the directory; these properties appear in lowercase letters in the *Flags* column.

Table 11-8

File Control Property

Property	I-Flags	Flags
Locked	L	l
Convenient Stage-out	C	c
Keep	K	k

The *I-flags* column also displays the residence priority integer, which is a value of 1–31 (set by root) and 32–63 (set by ordinary users).

The *Flags* column displays the file or directory's own properties as the corresponding lowercase letters and priority integer. A minus sign (–) indicates that the property is not set.

You can change file control properties with **emchmod**. The **emchmod** command sets the file or directory's staging control properties. In order to change properties you must be either the superuser or the owner of the file. Unlike **chmod**, **emchmod** clears properties if they are not specified on the command line.

The following example shows how properties are inherited:

1. Assign convenient property and residence priority.
`edm# mkdir archive`
`edm# emchmod -C -P36 archive`
2. Display properties.
`edm# emls -l archive`

```
Mag KB Stg KB I-flags  Flags  Staging media Volume Barcodes Filename
    1      0 --C- 36   ---- 0   -                archive
```

3. Create subdirectories.

```
edm# cd archive
edm# mkdir arc1
edm# mkdir arc2
```

4. Display properties.

```
edm# emls -l *
```

Mag	KB	Stg	KB	I-flags	Flags	Staging media	Volume	Barcodes	Filename
1		0	--C-	36	----	0	-		arc1
1		0	--C-	36	----	0	-		arc2

5. Create files.

```
edm# cd arc1
edm# touch file1
edm# touch file2
```

6. Display properties.

```
edm# emls -l *
```

Mag	KB	Stg	KB	I-flags	Flags	Staging media	Volume	Barcodes	Filename
0		0	----	0	--c-	36	-		file1
0		0	----	0	--c-	36	-		file2

emchmod has several optional switches that allow you to expand the HSM file control properties. By default, symbolic links are not followed by **emchmod**. If you use the **-s** option, the change applies to all the symbolically linked files.

If you know when you create a directory what properties all the associated files should have, specify the inheritable properties at that time. If you decide *after* you have created a directory what properties it should have, use the **-r** option, to recursively apply the properties.

Normally, **emchmod** silently sets and or changes properties. If you use the **-v** option, **emchmod** prints a message to your screen as it changes every file.

Compaction of Staging Media

Over time, some files that were staged out are deleted. Other files are staged back in, and some of them are modified. The old staged image of a deleted or modified file is considered *stale*.

Gradually, the number of stale files on staging volumes grows, and the volumes become candidates for compaction. When you compact staging volumes, you actually stage in the files that are not stale to magnetic disk and then, if they were not accessed recently, you stage them out again to new staging volumes.

Compaction is in effect a garbage collection process that creates space for new files by reducing the number of active staging volumes. It frees space in a full library unit for new staging volumes and/or ensures a pool of available media.

In most cases, staging media is compacted automatically via an **emcompact** entry in root's crontab file. With automatic compaction, **emcompact** automatically determines which volumes to compact.

You can also compact staging volumes manually, if you want to compact any additional volumes. Both automatic compaction and manual compaction use the **emcompact** command.

If you need to compact some staging volumes manually:

1. Use **dbreport**'s compaction report to decide which volumes to compact.

dbreport compaction

The compaction report is divided into three sections. The most likely volumes to compact are those listed in the last few lines of the first section. These are the volumes with the highest percentage of stale files.

2. Use **emcompact** to compact the volumes. In the case of EOs, which have two sides, you need to specify each side, or volume, separately. You can specify volumes by:
 - volume ID

- sequence number (for single-sided media)
- sequence number and side (for double-sided media)
- barcode (for single-sided media)

The following example compacts both sides of disk #10:

emcompact EO 10-1 10-2

You can type the command without any arguments to find out the legal media types. (In the case of tapes, you can specify a barcode.)

You can override EDM Migration's file residence policy by running **emcompact** with the **-p** (policy) option. The **-p** option ensures that all files from the compaction source volume are staged out to the compaction output volume and none remain on magnetic disks.

Administering Compaction

You can perform several tasks on a regular basis to ensure that compaction is working smoothly.

CAUTION: In order to ensure a complete file recovery process, you should disable automatic compaction and emvck as soon as you realize that you've lost a filesystem or a significant portion of one.

- Check the Volume Request window every morning to see if automatic compaction has blocked.
- Keep a supply of blank, easily accessible and unlabeled volumes, which you can label and allocate as compaction output volumes.

To increase the likelihood of maintaining a supply of free volumes, you can also convert unrestricted staging trails to restricted ones, or prelabel volumes for a specific trail.

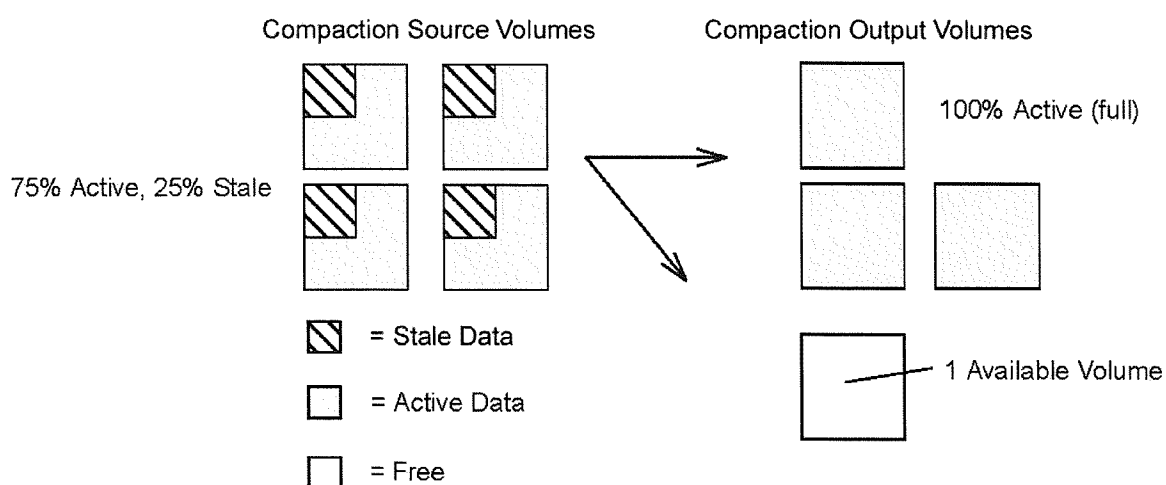
- Check the message files every day.

If automatic compaction frequently fails to reach the free goal, the system might have reached the limit of its data storage, given the number of volumes in the library unit or the time available for compaction.

In such a case, use the **dbreport appl_usage** report to determine whether enough stale space is available to reclaim, or whether you should add more volumes to your library unit. For example, in an EO system with volumes that average 25% stale data, you have to compact four volumes in order to get a single free volume (refer to Figure 11-10). If this rate is unacceptable in your environment, add more blank volumes.

Figure 11-10

Freeing Volumes for Use



Compacting Baseline Media

Compacting baseline volumes is similar to compacting staging volumes, except you can only compact baseline volumes manually. Compacting baseline volumes causes all currently active data associated with the volume to be copied to the current compaction volume associated with the baseline trail. New baseline compaction volumes are allocated as necessary.

As a general goal, you should limit the number of active baseline volumes to the number of active staging volumes. If possible, you should also limit the number of active baseline volumes to the number of slots in your library units. This facilitates recovery in the event of a site disaster by ensuring that all of the baseline media (to which active files are attached) fit in your library units.

To compact baseline volumes:

1. Run **dbreport baseline** weekly and refer to the “pct_stal” column for baseline volumes.
dbreport baseline
2. Use **emcompact** to compact the N most stale optical disks or tapes. (In the case of optical disks, there are two volumes per disk.) The value of N varies, but it is at least the number of active baseline media minus the number of active staging media or slots in your library units, whichever is less.

Therefore, if you have 55 active baseline media, 45 active staging media, and a 50-slot library unit, compact at minimum the ten most stale baseline media.

This results in a set of compacted baseline volumes. Note that these volumes are not deallocated and available for use until all existing baseline-relative backups that reference them expire.

Migration Reports

EDM HSM Option and EDM Migration contain several reporting tools that you can use to monitor system performance.

The **dbreport compaction** command generates three reports that you can use to determine which staging volumes to compact with the **emcompact** command.

The **emfsreport** program provides virtual filesystem statistics for the server and for network clients. It displays the amount of stageable and unstageable filesystem data, and the amount of data currently staged. Most importantly, it shows you the number of days worth of data being cached on magnetic disk. See “emfsreport and the Working Set” below for further details.

The **emsstat** program displays activity levels for the network migration server. It gets its information by accessing statistics that are kept in a shared memory segment that all active EDM Migration server daemon (emsd) processes use, or from a statistics file if emsd is not running.

The **emcheck** program checks the current migration configuration on the server or on a network client.

Refer to the appropriate man page for further details.

emfsreport and the Working Set

In a virtual filesystem, the files that you use most often, your *working set*, should fit on the magnetic portion of the filesystem, thus guaranteeing that most file accesses do not require staging volumes to be mounted. Your working set is often measured as the number of days worth of files that are stored on the magnetic disk. This is called the *working-set-in-days*. The ideal working-set-in-days varies from site to site and from filesystem to filesystem, but EMC recommends you try to maintain several days worth of data on your local magnetic disks.

Note that in the case of an archival environment where you expect to have no working set on your local disk, all accesses result in stage-in faults.

If your working set of files is considerably larger than your actual magnetic disk space, you experience system performance degradation. If you access large amounts of data within a short period, migration must stage files in and out continuously. This constant staging of files is called thrashing and should be avoided.

To find out your working set size and working-set-in-days, use **emfsreport**. The **emfsreport** program provides you with filesystem reports that enable you to monitor usage and thus fine tune your system. The **emfsreport** program can provide you with information such as:

- the number of files in a virtual filesystem
- the amount of stageable data in a virtual filesystem
- a filesystem's usage pattern for a particular day, for example, the total amount of space used by files created or modified in a day (that is, what is required for the green zone)
- the size of your working set

When you run **emfsreport** with the **-hva** option, you see a report of the virtual space by age since the last file access or modification. It also reports the size of the filesystem's working set. The working set *size* is the amount of stageable magnetic information that the filesystem can have without exceeding the LWM. The *days worth* value, which is based on observed access patterns, is the number of days it takes EDM Migration to cycle through an amount of data equal to the working set size.

Remember that this report is a snapshot of a specific moment in time. If you were suddenly to request many more files than normal, or if you were to create a large amount of new data, the working set period would be less than this estimate.

If you find that your working set is much larger than your physical space, that is, you have too few days worth of space, delete files or move them to another filesystem. Also, check that your system activity is evenly balanced across your disks. If your working set is still too large, add more magnetic disks or modify your application.

Baseline Backup

Baseline backup provides a highly efficient means of backing up large amounts of data. With baseline backups, you back up all of your most stable files, which, at minimum, consist of all the files that are staged out to the staging media. From that point on, you perform backups *relative* to the baseline; that is, the baseline backups take care of the data that is staged out, while the regular backups take care of everything else.

Baseline backup actually uses HSM software, rather than backup, to move data. In essence, it causes data to be staged out twice, and thus provides you with additional protection against the loss of your data. If, for example, you lose your primary staging media, due to fire or accident, you can still locate your files on the secondary staging media (that is, the baseline backups).

Restaging Data

HSM supports multi-level staging with its **restage** command, which incorporates enhanced **find** syntax. Using **restage** you can qualify files to stage and then migrate, or re-migrate files to a specified staging trail, force migration of a set of files, or establish an arbitrarily layered, staging hierarchy.

Multi-level staging is particularly useful when you want to free up space on your staging media. You can configure staging to automatically migrate data from magnetic disk to a staging trail, and then **restage** to another trail. If you want, you can move the restaged data to offline storage. See the **restage** man pages for more information.

Backup Completeness

Backup work items for filesystems that are under migration control have a completeness setting that prevents duplicate backups of the file data. The completeness setting limits the files for which the data portion is written to the backup. (The *extended inode* is included for each file scanned, regardless of whether its data is written out.)

You should leave the completeness settings at their defaults, listed in Table 11-9. The initial setting varies depending on the type of file being backed up (as noted in the table).

If you really need to change a completeness setting, you must edit `eb.cfg` directly; no setting is available from the graphical user interface. Editing `eb.cfg` directly is always a dangerous thing to do, so you should make a copy of your `eb.cfg` before editing.

Table 11-9

Completeness Settings

Setting	Description	Applicable For	Default For
All files	Back up the data portion of all files in the filespec, regardless of where they're stored and whether they're baselined.	All clients (this is the only option available for backup clients that are not also EDM Migration clients)	Non-migration clients and backup's database files on the server
Resident files only ¹	Back up the data portion for only those files that are resident (local to) the client; or, for the server, that are stored on the magnetic disk.	EDM Migration clients Levels 0-9 on the backup server (i.e., the local client)	—

Table 11-9 Completeness Settings (Continued)

Setting	Description	Applicable For	Default For
Files not backed up in migration store	If a file has been staged, only back up the data portion of the file if its staged version hasn't been backed up yet on the EDM Migration server.	EDM Migration clients	EDM Migration clients
Non-baselined files only	Back up the data portion for only those files that aren't baselined (for use after a baseline is taken). This option is only available if you have Baseline Backup.	Local (server) client (but not backup's database files)	Local client (except backup's database files)

1. EMC recommends that you use **Files not backed up in migration store** with EDM Migration network clients, and **Non-baselined files only** for the local migration server. The **Resident files only** setting can leave you vulnerable unless there is a backup of the client store. If you don't back up a file that has been staged out, but you lose the file's client store on the server before the server's files are backed up, the only way you will be able to recover the file is from an old backup.

12

How Migration Works

This chapter describes the roles of the HSM daemons, processes, and database files in migration services on the server and clients.

The following topics are discussed in this chapter:

- What Happens When You Enable Migration
- How Stage-Out Works
- How Stage-In Works
- How the User-Level Commands Work
- How the Network Migration Server Works
- How Compaction Works

What Happens When You Enable Migration

When you enable filesystem migration, you set up migration parameters, such as watermarks and a delay factor, and you specify the type of media to which files migrate.

When you enable filesystem migration, HSM does the following:

- It stores configuration information in the HSM configuration database.
- It creates a holding place for the migration candidate list (See “Candidate List Generation” on page 12-5).

Migration Configuration Database

The HSM server and every HSM client contains a migration configuration database. This database consists of structured text files that are updated by the **emstconf**, **emfsconf**, and **emsysconf** commands and by the functions you perform when using the HSM Configuration Interface.

CAUTION: Although these files are text files, you should never attempt to modify them with an ordinary editor. The configuration commands and the HSM Configuration interface do more than just modify the files; they also know how to interact correctly with any staging processes that are running.

The database contains information that specifies which filesystems are stageable, when files should be staged, and which staging templates filesystems are assigned to.

On client systems, the database also lists the fileserver and store that staging templates are assigned to.

The database files are stored in the `/usr/epoch/etc/mal/` directory.

Figure 12-1

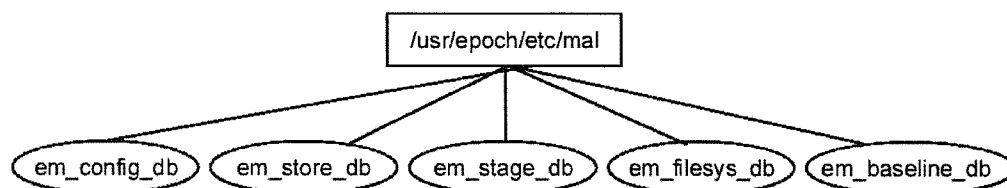
Configuration Database

Table 12-1 lists the database files.

Table 12-1

Migration Database Files

Argument	Description
em_config_db	Text file containing system-wide configuration data. You update this file with the emsysconf command or when you change global properties with the HSM Configuration interface.
em_stage_db	Text file containing staging template configuration data. You update this file with the emstconf command or when you change staging template information with the HSM Configuration interface.
em_store_db	Text file containing client store information. You update this file with the emstconf command or when you change store information with the HSM Configuration interface.
em_filesys_db	Text file containing per-filesystem configuration data. You update this file when you issue the emfsconf command or when you change filesystem information with the HSM Configuration interface.
em_baseline_db	Text file containing baseline backup information.

How Stage-Out Works

HSM stages out files during nightly staging runs (periodic staging), when disk space usage crosses the high watermark (demand staging), or when the **emstage** command is issued. (See “When Files Stage In and Out” on page 11-2 for further details.)

There is a limitation of a maximum of 2GB minus 1KB for the size of files to be staged out or staged in.

Both periodic and demand stage-outs occur via the interaction of the following daemons and processes:

- The EDM Migration file monitor daemon (**emfmd**)
- The **emmasterd** daemon
- The **em_make_cl** process

Both the **emfmd** and the **emmasterd** daemon are started at boot time. The **emfmd** detects high watermark faults and then communicates with the **emmasterd** daemon, which starts a worker daemon. The worker daemon starts up **em_make_cl**, which fills in the candidate list and thus, determines which files to stage out.

These daemons and processes are described more fully on the following pages. See “How the User-Level Commands Work” on page 12-7 for information about **emstage**.

The File Monitor Daemon (**emfmd**)

The **emfmd** detects events that require migration intervention and then communicates with the **emmasterd** daemon. The **emmasterd** starts a worker daemon which actually stages out the file(s). (Previous versions of HSM provided the functions of the **emfmd** via the Unix kernel.)

The **emfmd** detects the following types of events:

- Filesystem space utilization at (or above) the high watermark

- Read or write accesses to prestaged or staged files
- Deletions of prestaged or staged files
- Filesystem mount and unmount operations

When a user program requests a file that is not staged out, the **emfmd** determines that no staging actions are necessary and normal system processing proceeds. The **emmasterd** daemon only gets involved when it is necessary to stage out a file.

The Master Staging Daemon (emmasterd)

When an HSM server or a migration client boots, it starts the master staging daemon, **emmasterd**, from /etc/rc3.d/S21mal. This daemon is responsible for staging out files from the clients to the server and from the server to the staging media. The **emmasterd** daemon keeps disk space utilization below the high watermark by staging out files and releasing their magnetic blocks.

Thereafter, the master starts, monitors, and restarts one worker daemon per filesystem, both periodically and on demand, when the **emfmd** notifies it that filesystem utilization exceeds the high watermark. The workers are also named **emmasterd**.

Only one real **emmasterd** process can ever run – the worker processes are simply forked copies. They appear as **emmasterd** processes when you run the **ps** command.

The simplest way to tell the difference between a worker process and the real **emmasterd** process is to look at the /usr/epoch/etc/mal/emmasterd.pid file. When **emmasterd** first begins execution, it writes its process ID into this file.

Candidate List Generation

When migration needs to stage files, an **emmasterd** worker process spawns **em_make_cl**, which creates a prioritized list of stageable files.

In selecting files to stage out, **em_make_cl** evaluates the time since the last file access, the size of the file, and the file's residence priority attribute (see the man page for **emchmod -p**).

Files with lower residence priority are usually staged first. (The lowest priority is 63; the highest priority is 1.) Thus, files with priorities from 33 to 63 are more likely to be staged out, and files with priorities from 0 to 31 are less likely to be staged out.

Only the superuser can raise priorities (by setting priorities in the range 1–31). All users can lower priorities.

What Happens When a File is Staged Out

The first time a file is staged out, migration writes the file's entire magnetic image to the next level in the staging hierarchy, that is, to the client store or the staging media. It keeps a small portion of the file on magnetic disk and releases the rest of the magnetic space. The portion of a staged-out file that remains on magnetic disk is called the *fencepost*.

This fencepost is useful, because many commands, such as **file** and **head**, only need to read this small portion of the file. Consequently, when these commands are run, HSM doesn't need to stage in the entire file. When a file is staged out a second time, it releases the magnetic space occupied by the fencepost.

Files that are staged in reside on both magnetic disk and the staging media (or client store). If the file is staged out again without being modified, migration uses the same staged image and releases the magnetic space. If the file is modified and then staged out, migration writes a new image on the staging media or client store.

How Stage-In Works

Stage-ins occur due to the interaction of the HSM file monitor daemon (**emfmd**) and the stage-in daemon (**emsid**). The **emfmd** detects a request for a staged out file (or a request to delete a staged out file) and notifies the **emsid**, which stages in some portion of the file, or, in the case of delete operations, deletes the file's staged image.

Scripts that run at system boot time automatically start up several stage-in daemons. Each stage-in daemon can handle one stage-in request at a time, which allows for multiple, simultaneous stage-in requests.

A staged-out file is logically divided into small chunks, called *buckets*. A file can be divided into anywhere from one up to a maximum of 64 buckets. The size of the buckets is based on the file's size. As the size of the file increases, the size of the bucket also increases.

When certain applications and processes request to read a portion of a staged-out file, migration stages in only those buckets that contain the requested data. Whenever a file is modified in any way, migration stages in the entire file and makes the previous staged image invalid or *stale*.

How the User-Level Commands Work

The user-level commands (**emchmod**, **emls**, **emstage**, and **embisi**) enable users to set and list file attributes for stageable filesystems and to specifically request the staging of particular files. The user-level commands interact with the migration RPC daemon (**emrpcd**), which, in turn, interacts with the **emfmd**.

For more information about the commands, see the man page for each individual command.

How the Network Migration Server Works

Network migration server software runs on the EDM server and consists of the following components as listed in Table 12-2.

Table 12-2

Components of the Network Migration Server

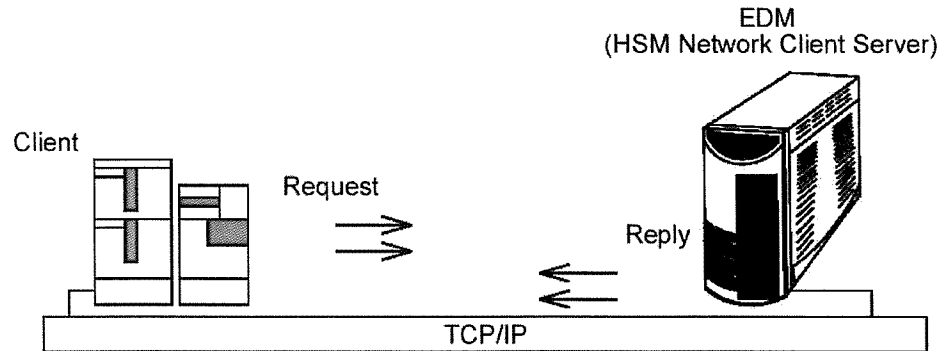
Component	Description
EDM Migration protocol	The communication between server and clients
EDM Migration daemons	Daemons that service client requests
Network migration server database	Files that track network migration activity and the default client store values
Client stores	Directories that hold client bitfiles

The EDM Migration Protocol

Network migration server software uses the EDM Migration protocol to enable communication between the clients and the server. The EDM Migration protocol is a remote procedure call (RPC) protocol that consists of pairs of request and reply messages that are passed between the client and server.

Figure 12-2

EDM Migration Protocol



The EDM Migration protocol is based on a connection-oriented transport protocol (TCP/IP), requiring each client to establish one or more virtual circuit connections to the server. This protocol reduces the effects of transport latency (round trip time) on performance and permits network migration to function well over both local and wide area networks.

Network Migration Server Daemons

Network migration server activities are carried out by a hierarchy of daemons (all named the EDM Migration Server Daemon, **emsd**) and controlled by a set of administrative commands. (See the appropriate man pages.)

The **emsd** is responsible for the following activities:

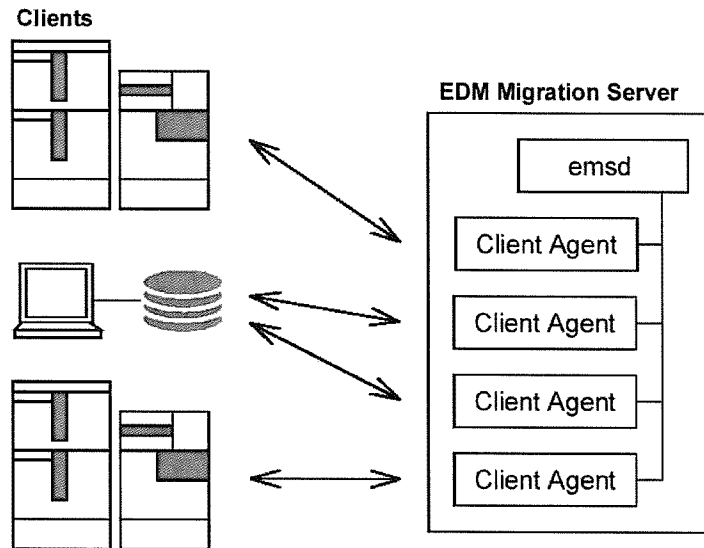
- initializing the global statistics shared memory segment
- parsing the global configuration file and store database
- parsing the store configuration files for each client store
- registering Network RPC service information
- listening for client system connection requests

The **emsd** daemon is started at boot time. When client connection requests arrive, **emsd** spawns subprocesses called *client agents* to handle them. The **emsd** creates a client agent process for each connection.

The client agent handles all requests over that connection for the lifetime of the connection.

Figure 12-3

Server Daemons



When an agent receives a request to access bitfiles in a particular store, the agent looks up the store configuration and state information in data structures inherited from the emsd process. A client agent can access any store to which its client system has access permission.

Network Migration Server Database

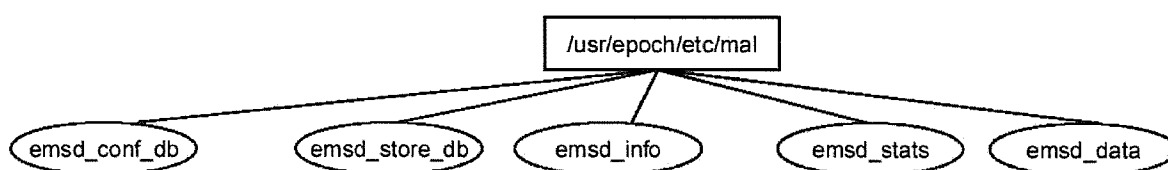
The network migration server has a global configuration database that contains information about network migration activity and the default client store values. The global configuration database files reside in /usr/epoch/etc/mal.

Although both the `emsd_conf_db` and the `emsd_store_db` files contain editable text descriptions of the configuration, do not edit these files directly. Instead use the server's configuration commands or the Configuration Interface to make any modifications to the database.

CAUTION: Editing these files directly may result in loss of data.

Figure 12-4

Global Database Files



There are five database files as described in Table 12-3.

Table 12-3

Global Database Files

File	Description
<code>emsd_conf_db</code>	Text file that defines the limits on the EDM Migration protocol requests and the default values for client store configurations.
<code>emsd_info</code>	Binary file that contains information about the currently executing <code>emsd</code> process.
<code>emsd_store_db</code>	Text file that contains a list of configured client stores and their locations on the server.
<code>emsd_stats</code>	Binary file that contains cumulative statistics on EDM Migration protocol traffic and client agent activity.
<code>emsd_data</code>	Binary file that contains the EDM Migration usage history on the server.

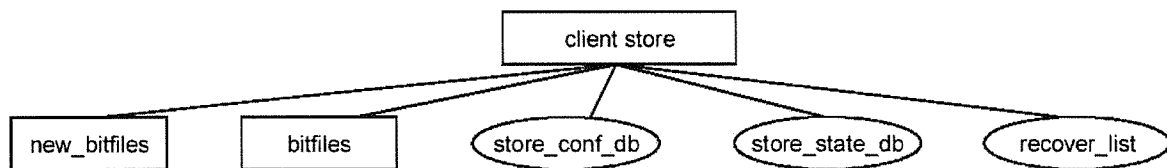
Client Stores

Each client store has its own file hierarchy and is logically independent from every other client store. The client store's top-level directory contains three files and two subdirectories.

As system administrator you see these files and directories when you list the contents of the client store directory.

Figure 12-5

Client Store Organization



The client store's top-level files and directories are listed in Table 12-4.

Table 12-4

Client Store Files and Directories

File/Directory	Description
store_conf_db	Text file of the store's configuration information.
store_state_db	Text file of the store's state information that the client agent keeps current.
recover_list	List of the bitfiles to restore from the server's backups.
new_bitfiles	Temporary holding directory for bitfiles that are being created as part of a stage out from a client system.
bitfiles	Directory that contains the completed bitfiles in a 3-level hierarchy.

When the client agent creates a bitfile, it gives it a 16-digit hexadecimal name and places it in the new_bitfiles directory. The bitfile remains there until it is completely written. Once the bitfile is complete, the client agent moves it to the bitfiles directory.

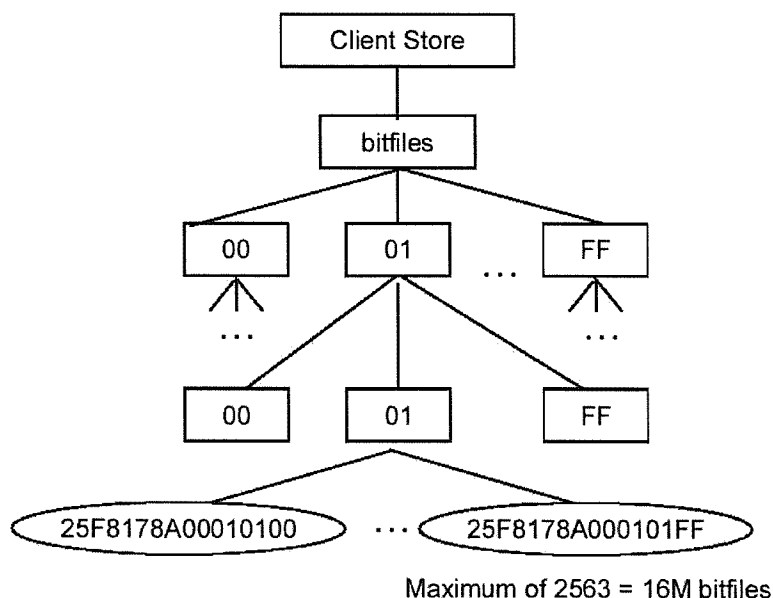
Bitfiles

Bitfiles are stored at the bottom layer of a directory hierarchy as shown in Figure 12-6. Bitfile names are 16-digit hexadecimal numbers representing the lower 64-bits of a file's bitfile ID. (The bitfile ID consists of the bitfile name plus the store ID.)

Migration uses this organization so that a bitfile can be located by using the hexadecimal encoding of the bitfile ID.

Figure 12-6

Bitfile Hierarchy



How Compaction Works

Compaction is in effect a garbage collection process that creates space for new files by reducing the number of active staging volumes. On systems with erasable staging media, compaction occurs automatically via an **emcompact -c** entry in root's crontab file.

Compaction Goals

The object of automatic compaction is to reclaim storage space on the staging media by making sure that each staging trail has at least a certain number of available volumes to stage to. By always maintaining a sufficient pool of volumes, minimal operator intervention is needed.

The **emcompact** command operates, however, under a pair of competing goals:

- To compact as many volumes as possible without blocking itself. (A block occurs when EDM Migration needs to allocate a new volume in order to compact a volume, but there are no new volumes available.) The **emcompact** program will block while waiting for a new volume, and a new volume request will be posted to the Volume Request window.
- To begin compaction with the trail that needs it the most, that is, the trail with the fewest number of available volumes.

The **emcompact** program only considers the volumes in the library units to determine the number of volumes available for allocation to each staging trail. If that number is less than the value specified in the **-a** (automatic) option, it examines all the volumes and compacts enough to provide an adequate number of free volumes. By default, **emcompact** ensures there are at least three free volumes for each staging trail. After the source volumes are compacted, they are erased and made available for reuse.

Example

In the following crontab entry, **emcompact** is set to run at 1:00 a.m. every morning:

```
00 1 * * * PATH=/usr/epoch/bin:$PATH;export PATH;emcompact -c >/dev/null 2>&1
```

This command specifies that autocompaction should be done according to the directives contained in the autocompaction configuration file `/usr/epoch/etc/mal/emcompact.cfg`. This form is typically called to compact reusable volumes in a library unit. See `cron(1m)` and the `/usr/lib/crontab` and `/usr/epoch/etc/mal/emcompact.cfg` files distributed with the system.

The Compaction Process

When you run compaction automatically, **emcompact** first chooses a staging trail and then decides which volumes to compact within that trail. For each staging trail, **emcompact** can allocate, as compaction output volumes, any volume that is already allocated to that staging trail or any unrestricted volume from another staging trail, as long as both sides are available.

Table 12-5 shows some sample staging trails and the volumes available for each trail at a certain point in time. This information is used in the process that is described below.

Table 12-5

Available Compaction Volumes

Staging Trail	Available Volumes
Engineering	6
Engineering_archive	4
Documentation	2
Documentation_archive	2
CAD	1
CAD-archive	0

The process is as follows:

1. First, **emcompact** looks for trails that have less than *n* available volumes. (*n* is the number specified with the **-a** option, 3 in this example). Automatic compaction only operates on those trails that have less than *n* available volumes, so **emcompact** selects only the Documentation, Documentation_archive, CAD, and CAD_archive staging trails.
2. Then, **emcompact** selects the trails that are least likely to cause a blocked process and require operator intervention. As such, it first looks for a trail with at least two available

volumes, in this case, Documentation and Documentation_archive. If there's more than one trail with at least two available volumes, but less than n available volumes, **emcompact** first chooses the trail with the fewest available volumes.

3. Then, **emcompact** selects the piece of media that is the most stale, taking into consideration *both* volumes, in the case of an optical disk. The **emcompact** program also takes into consideration the disks' availability, so that any disks that are restricted to other trails cannot be considered.
4. On that disk, **emcompact** selects the stalest side and compacts it.
5. The **emcompact** program repeats Step 2 through Step 5 until all the trails that started with at least 2 available volumes have had volumes freed up and now have at least n available volumes.
6. The **emcompact** program repeats Steps 2 through Step 5 until all the trails that started with 1 available volume (CAD) have at least n available volumes. At this point there is a greater potential for a blocked process.
7. The **emcompact** program repeats Steps 2 through Step 5 until all the trails that started with 0 available volumes have at least n available volumes. At this point there is the greatest potential for a blocked process.

At any time in this sequence, **emcompact** can run out of time, depending on the length of time specified in the **-e** switch. For example, if the command line specifies **-e 120**, **emcompact** will terminate in two hours. In this case, the program will exit. The next time compaction runs it starts the selection process over again. Most likely, it decides that the volume it was in the process of compacting is the best candidate to compact.

How Long Compaction Takes

Compaction takes a considerable amount of time; up to a few hours is not unusual. **emcompact** requires about five minutes to scan each filesystem and identify active files. The time required to stage the files in and out again depends on the number of blocks and can increase if compaction triggers event-driven staging. During compaction, all resident and staged-out files (including the files staged out to the volumes being compacted), in all filesystems, can be used normally. If compaction is interrupted, you can restart it on the same volumes.

Baseline Compaction

Compacting baseline volumes is similar to compacting staging volumes. Compacting baseline volumes causes all currently active data associated with the volume to be copied to the current compaction volume associated with the baseline trail. New baseline compaction volumes are allocated as necessary.

When that baseline volume is compacted it cannot be immediately reused since there may still be baseline-relative backups referencing the volume. When you compact a baseline volume you move the data of all active files that reference this volume to a different baseline volume. This means that no new baseline-relative backups reference this baseline volume.

A compacted baseline or staging volume has no active data. You can verify this by checking the volume's **Used 1k blocks** field (for an optical volume) or **Used files** field (for DLT) with the EDM Library Unit Manager window.

Deallocation and Reuse

The deallocation and reuse of baseline volumes is handled by EDM Backup. When expiring backups, EDM Backup checks for active data on all baseline volumes by checking each volume's "KB used" field. If the field is zero, either due to the volume having been compacted or the volume having grown completely stale, EDM Backup considers this volume for deallo-

cation. After all of the baseline-relative backups that reference these baseline volumes expire, **ebexpire** can deallocate the baseline volume, making it available for reuse.

Note that you can compact any baseline volume at any time. The deallocation of the volume is handled by EDM Backup, which knows not to deallocate it if it is still required by any baseline-relative backup.

Active Baseline Volumes

A baseline volume is considered “active” if it has data on it that is needed by an unexpired baseline-relative backup. Such a baseline volume cannot be reused until it has been compacted *and* deallocated.

Recovering from Site Disasters

You should limit the number of active baseline media to the number of active staging media, and you should make sure that you can fit all of your active baseline media into your library unit at one time.

The reason for this is that in the case of a site disaster you need to replace your damaged staging media with your baseline media. (Note, however, that although this should be a maintenance goal, there is no real relationship between staging and baseline media.)

Furthermore, in the case of a site disaster, the fewer baseline volumes you need to deal with, the better. That is, you do not want to have to purchase an extra library unit because you have more baseline media than staging media.

See “Compacting Baseline Media” on page 11-29 for information on how to compact baseline volumes.

13 HSM Command-line Tasks

Generally, HSM is configured from the graphical interface and requires very little day-to-day maintenance. This chapter describes the non-routine configuration and maintenance tasks that are performed mostly from the command line.

- HSM Commands
- Test Staging
- Set Up Periodic Staging
- Tuning for Staging to Tape
- Coordinating Automatic Procedures
- Working with Individual Files
- Checking the Staging Configuration
- Copying and Moving Data
- Monitoring Storage Space and File Sizes
- Maintaining Non-Stageable Filesystems
- Managing Your Magnetic Disks
- Populating Filesystems
- Disabling Filesystem Staging
- Compacting Staging Media

- Compacting Baseline Media
- Clearing Incomplete Bitfiles
- Gathering Migration Store Statistics
- Checking a Network Client's Staging Configuration
- Troubleshooting HSM
- Restoring a Lost or Damaged Staging Volume
- Restoring a Lost or Damaged Staging Trail
- Restoring a Lost or Damaged Filesystem

HSM Commands

A brief description of the administrative and configuration commands available for local and network HSM systems can be found in "HSM Man Pages" on page 18-9.

Test Staging

Test that staging works by copying a large file, such as `/etc/termcap`, to a stageable filesystem, stage it out, and verify that staging took place. Refer to the corresponding man pages for details concerning command arguments.

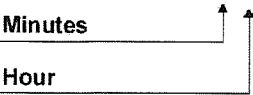
```
emc# cd /home1
emc# cp /etc/termcap
emc# emstage termcap
emc# emls
```

Mag	KB	Stg	KB	Staging media	Filename
	8		0		lost+found
	8		87	#0155-a doc	termcap

Set Up Periodic Staging

Make sure an entry exists in root's crontab file for scheduling nightly, periodic staging runs. The following example is inserted into root's crontab file by the HSM installation procedure.

```
00 0 * * * /bin/kill -HUP `cat /usr/epoch/etc/mal/emmasterd.pid` >/dev/null 2>&1
```



Tuning for Staging to Tape

EDM with HSM option comes configured for staging to optical media. If you are staging to tape, you can improve performance by causing larger stage-ins, and thereby less tape repositioning.

You do this by using two parameters in the server's `/usr/epoch/etc/msl/msl.cfg` file and then enabling this modified configuration file.

Optical media has a relatively quick *seek time* when compared to tape. Therefore, a small amount of data can be efficiently staged in from optical. This is how HSM is configured by default.

For tape, however, reading small amounts of data for multiple users can cause thrashing against the tape drive. Larger stage-ins are required to cause less tape repositioning. The larger stage-ins are possible because magnetic media has a faster transfer rate than optical and a much greater storage density.

Stage-to-Tape Tuning Parameters

The parameters are:

- `MSP_READ_AHEAD_PERCENTAGE`

The percentage of the file to be read on each read-ahead during a stage-in. It must be a whole number.

- `MSP_READ_AHEAD_MAX`

A throttle on `MSP_READ_AHEAD_PERCENTAGE`, it is the maximum number of bytes that are read on any give read-ahead. (This parameter is used to prevent the stage-in of a large file from taking over the system.)

The default settings for optical are:

```
MSP_READ_AHEAD_PERCENTAGE=25
MSP_READ_AHEAD_MAX=1048576
```

If you are staging in from magnetic tape and have concurrent users you may want to change these to:

```
MSP_READ_AHEAD_PERCENTAGE=34
MSP_READ_AHEAD_MAX=62914560
```

For example, if a user tries to read in a 1 gigabyte file and the MSP_READ_AHEAD_PERCENTAGE is 34, then HSM tries to read in 300 megabytes.

If you set MSP_READ_AHEAD_MAX=62914560, the stage-in is limited to 60 megabytes (or about 1 minute of time).

Stage-to-Tape Tuning Procedure

To edit these parameters and enable the modified configuration file:

1. Edit the /usr/epoch/etc/msl/msl.cfg file to edit or add the two parameter values.

Note: You should make a copy of the original /usr/epoch/etc/msl/msl.cfg file.

2. Kill an emfmd daemon (killing one child process should kill them all).
3. Do an **init 6**

or

Restart the emfmd daemons with:

```
/usr/epoch/lib/msl/emfmd
```

Coordinating Automatic Procedures

To ensure maximum efficiency, you should coordinate all of the automatic backup and staging procedures that are run through root's crontab file. Table 13-1 shows two sets of recommended

order of tasks: one for systems with only local HSM and backup and the other for systems with local and network HSM and backup.

Table 13-1

Recommended Order of Procedures in crontab

**System with local HSM
and backup**

1. Server: **emvck**
2. Server: periodic stage out
3. Server: compaction
4. Server: baseline backup (optional)
5. Server: regular backup

**System with local and network
HSM and backup**

1. Client: periodic stage out
 2. Server: **emvck**
 3. Server: periodic stage out
 4. Server: compaction
 5. Server: baseline backup (optional)
 6. Server: regular backup
 7. Client: backup
 8. Server: database backup
-

The remainder of this section provides additional detail for these activities.

Backup automatically performs baseline backups before regular backups, and in sites with HSM clients, perform a database backup last. You only need to schedule client backups (step 7) after the server backup (step 6) in sites with HSM clients.

Client (Periodic Stage Out)

It is always best to stagger your network client's periodic staging runs so that their activity does not overload the network or the server. If, however, you stage multiple clients simultaneously, performance improves if the clients stage to different server disks.

Note: You should configure a larger delay factor (closer to twenty minutes) for older and slower hardware with large files or a large numbers of files.

If a client system has more than one stageable filesystem, stagger the staging of each filesystem by 12 to 20 minutes. It is especially important to set this delay factor on filesystems that contain large files of more than one MB. Running more than one staging operation simultaneously can degrade throughput.

The following entry in root's crontab file starts a new staging day at 12:15 every night:

```
15 0 * * * /bin/kill -HUP `cat /usr/epoch/etc/mal/emmasterd.pid` > /dev/null 2>&1
```

Refer to “Periodic Staging and Filesystem Delay” on page 11-22 for further information.

Server (emvck)

The **emvck** (volume check) program reads filesystem information on magnetic disk and compares it to the database. If the results for a staging volume do not match, it generates accurate counts, updates the database, and logs a message. The following entry in root's crontab runs **emvck** at 11:45 every night:

```
15 23 * * * PATH=/usr/epoch/bin:$PATH;export PATH;emvck >/dev/null 2>&1
```

Server (Periodic Stage Out)

Schedule nightly periodic staging of all client stores. Stagger the staging so that the activity does not overload the optical library unit.

Server (Compaction)

Schedule nightly compaction of staging volumes by running **emcompact** every night from root's crontab file. After the source volumes are compacted, they are made available for reuse.

Check the message files every day. If automatic compaction frequently fails to reach the free goal, the system may have reached its limit, given the number of volumes in the library unit. Use the **dbreport compaction** report to determine whether there is enough stale space to reclaim, or whether you should add more disks to your library unit. Refer to “Compacting Staging Media” on page 13-28 for further details.

Server (Baseline Backup)

Baseline backups are run automatically before regular backups. The recommended baseline backup procedure is to have backup templates use both primary and alternate trailsets, with both trailsets specifying a single baseline trail.

You can back up the server, several clients, and the server database all within a single backup template.

Server and Client (Backup)

ebbackup should back up the server first and then your clients. If you are running network HSM, it is important to back up the client stores on the server, *before* you back up the clients. The following crontab entry starts a backup at 10:30 each night, using the backup template called *default*:

```
30 22 * * * /usr/epoch/EB/bin/ebbackup default > /dev/null 2>&1
```

Refer to Chapter 14 “Start of Backup and Related Processing” for more information.

Server (Backup Database)

Always back up the backup databases on the server after you back up the clients. This is also the case even if you have no network clients, because the server is considered a “local” client to the backup software. Database backups provide you with complete information about both the server and client backups and shorten disaster recovery time because they enable you to restore the database independently from the files that were already backed up.

By default, the backup software ensures that the backup database is backed up last.

Rotating Error Logs

Log files reside in a non-stageable filesystem in the `/var/adm/epoch` directory. To prevent the log files from growing excessively, the **epnewlog** script, which is run weekly from root's crontab file, will rotate or archive a log file to a stageable filesystem (`/usr/epoch/adm/rotated`) whenever the log file grows to more than one megabyte. The script rotates the concise log and the mntfault log using the usual Unix-style rotation scheme (`*.0, *.1, *.2,...`):

The script archives the detail log permanently to `/usr/epoch/adm/archived` using a date-based suffix.

epnewlog does the following:

1. Moves each message log file from the `/var/adm/epoch` directory to the `/usr/epoch/adm/rotated` directory. For example, the message log file `/var/adm/epoch/concise` is moved to `/usr/epoch/adm/rotated/concise`.
2. Makes each rotated log file in `/usr/epoch/adm` a version.0 file. For example, the message log file shown in step 1 becomes `/usr/epoch/adm/rotated/concise.0`. Each time **cron** runs the **epnewlog** script, the file suffix is incremented (`.1, .2, .3`) and finally deleted.
3. Assigns a date-based suffix to the detail log file in `/usr/epoch/adm/archived`.
4. Creates new message files in the `/var/adm` directory.
5. Restarts the **syslogd** process.

For the rotated log files, this procedure saves an entire month's worth of log data by rotating the log file names: `messages.0` → `messages.1` → `messages.2` → `messages.3` → deleted. Because the `messages.[0-3]` log files are in the `/usr` filesystem and are no

longer growing, they are now candidates for HSM to staging devices. Use a similar approach if you've configured any other log files.

Working with Individual Files

The HSM Configuration interface handles data among several layers in a storage hierarchy. However, the software also provides command-line tools for manipulating individual files. The following procedures describes these tools:

- Staging out files
- Staging in a set of files
- Tagging a set of files for future stage out
- Locking a file on magnetic disk

Staging Out Files

During normal system use HSM manages staging for you. There are times, however, when you know that a group of files is no longer needed. For example, when you finish a project, you can explicitly stage those files out.

Use the **emstage** command to stage out or prestage one or more files. The following example stages out the files, drawing1 and drawing2:

```
emc% emstage drawing1 drawing2
```

There are command line arguments that let you prestage files, to recursively descend subdirectories while staging, to follow symbolic links, to stage to a particular volume, and to stage to a volume that contains a particular file. See the **emstage** man page for details.

Staging In a Set of Files

If you know that a group of staged-out files are needed soon, you can use the **embsi** command to stage them in. The **embsi** command stages the files in and attempt to make them all resident on magnetic disk.

By default, **embsi** does not attempt to stage in files if there is not enough available space for them on magnetic disk. Also, by default, **embsi** does not update access times. To force **embsi** to stage in files even if not enough magnetic disk space is available, use the **-f** option; to update access times, use the **-a** option.

When access times are not updated, **embsi** estimates available free space; when access times are updated, **embsi** estimates the total disk space in the filesystem. If not enough space is available, **embsi** displays the amount required and its estimate of the amount available. You can select a smaller set of files to stage in, explicitly stage out files, or force the stage in.

Forcing a bulk stage in, when there is not enough space, almost certainly means that HSM will have to stage out some files. If the access times are updated, the chances are that many of the files that are being staged in remain on magnetic disk and other files are staged out. If the access times are not updated, the chances are that at least some of the files being staged in do not remain on magnetic disk.

The following example recursively stages in all the files in the current directory:

```
emc% embsi -r .
```

Tagging a Set of Files for Future Stage Out

Use **emchmod -C** to tag files so that they are staged out at the next convenient time, which is usually the next periodic staging run. Remember that, **emchmod**, unlike **chmod**, clears properties if they are not specified on the command line. Therefore, you should always use **emls -l** first to determine the properties that are already set. You must be superuser or the owner of the file(s) to change properties.

To tag a set of files for future stage out:

1. Change to the directory that contains the files you want to stage out.

```
emc# cd archive
```

2. Use **emls -l** to determine the properties that are already set.

```
emc# emls -l *
```

Mag	KB	Stg	KB	I-flags	Flags	Staging media	Volume barcodes	Filename
1024		0		----	0	1----	60	-
	24		898	----	0	----	0	#002-a Archive
	1		0	--CK	60	----	0	-
								filexyz
								fileabc
								dirabc

3. Use **emchmod** with the **-C** option to indicate that the specified files should be staged out at the next convenient time. (The **-P** option sets the residence priority.)

```
emc# emchmod -C -P36 filexyz
```

Locking a File on Magnetic Disk

Use **emchmod -l** to lock file(s) on magnetic disk and prevent the files from being staged out. Remember that, **emchmod**, unlike **chmod**, clears properties if they are not specified on the command line. Therefore, you should always use **emls -l** first to determine the properties that are already set. You must be superuser or the owner of the file(s) to change properties.

To lock a file on magnetic disk:

1. Change to the directory that contains the subdirectory or files you wish to lock on magnetic disk.

```
emc# cd archive
```

2. Use **emls -l** to determine the properties that are already set.

```
emc# emls -l *
```

Mag	KB	Stg	KB	I-flags	Flags	Staging media	Volume barcodes	Filename
1024		0		----	0	----	60	-
	24		898	----	0	----	0	#002-a Archive
	1		0	--CK	60	----	0	-
								filexyz
								fileabc
								dirabc

3. Use **emchmod** with the **-l** option to lock the specified files on magnetic disk.

```
emc# emchmod -l filexyz
```

Consider locking down VxFS reserved files. Reserved files are configured on VxFS to remain on the magnetic disk, so they are good candidates for locking. See the VxFS documentation for information on reserved files.

Locking too many files can seriously impact HSM performance. Before you lock files onto magnetic disk, try small changes to the residence priority (see the **emchmod** man page). Be careful about setting the inheritable lock property on a directory. All files and directories created below that point inherit the lock property.

Checking the Staging Configuration

Use the **emcheck** command to check the staging configuration. You must be superuser to run **emcheck**. If you type the command without any arguments it checks the configuration database, warns you of potential problems and corrects inconsistencies. If you use the **-v** switch (verbose) you see more information. If you use the **-r** switch (read-only), **emcheck** does not correct any problems that it may find.

The **emcheck** command checks the database in eight phases. The first four phases, D1-D4, verify the semantics and the syntax of the configuration database. The second four phases, S1-S4, verify the system as a whole.

The **emcheck** command displays the following lines:
beta# **/usr/epoch/bin/emcheck**

```
*** Phase D1 - Checking configuration database files
*** Phase D2 - Checking status of Epoch Migration servers
*** Phase D3 - Checking store database against server config
*** Phase D4 - Checking status of stores
*** Phase S1 - Check Epoch Migration directories
*** Phase S2 - Check for running Epoch Migration daemons
```

Copying and Moving Data

With HSM you have the ability to store vast amounts of data throughout your network. Often you may find the need to copy or move data from one location to another – between filesystems, network clients, or servers, for example. This section provides step-by-step instructions for copying and moving large amounts of data between various locations in your network. The topics are:

- Migrating data from one staging trail to another
- Copying files from one filesystem to another
- Moving and copying files between HSM systems
- Copying files between network HSM clients
- Copying files to a non-EDM system

Migrating Data from One Staging Trail to Another

After you've had your system for a while, you may want to archive some older data to tape or optical disk. To do this, use the **restage** command on the EDM server to migrate files from one staging trail to another.

1. Decide which staging trail you want to restage the data to.
2. If necessary, create a new staging template/trail with the **emstconf** command.
3. Use the **restage** command to migrate the data from the existing staging trail to the new one.

The following command moves staged files from the doc trail to the tape1_trail. The command moves only those files that haven't been modified or accessed in the last 30 days.

```
# restage -t tape1_trail -R /data1 -mtime +30 \ -atime +30 -staged_to_trail doc
```

See the **restage** man page for additional examples.

Copying Files from One Filesystem to Another

The fastest, most efficient way to move or copy large amounts of data from one filesystem to another is with the **ebcp** command. The **ebcp** command copies files from one place to another within an HSM-enabled system, between HSM-enabled systems, or from an HSM-enabled system to a non-HSM system.

The **ebcp** command automatically determines whether the destination filesystem is under HSM control. If so, **ebcp** copies the files that are not staged out to the new location. In addition, **ebcp** will create a filesystem entry for staged-out files and attach these entries to their staged images. If the destination location is not under HSM control, **ebcp** will copy everything, including the files that are staged out.

When copying to a filesystem that is not under HSM control, you must be sure that enough space is available to hold all of the files that are copied.

The following example copies all of the files in the /data1 filesystem to the /data2 filesystem on the same HSM-enabled system:

1. Change to the source directory:

```
emc# cd /data1
```

2. Use **ebcp** to copy the files:

```
emc# ebcp . /data2
```

After you copy the files, you can delete the originals.

Moving and Copying Files Between HSM Systems

There are two general approaches to copying files between HSM-enabled systems. Both approaches use the **ebcp** command. In the first approach, you simply copy all of the files and directories from one fileserver to another, including everything that is staged out, without transferring the staging media to the new fileserver.

In the second approach, you copy only the magnetic-resident data to the new fileserver and then reattach to the staging media. This second approach is significantly faster and is especially recommended when the quantity of data is too great to transfer over the network.

Copying Files to Another HSM System (No Media)

The following example copies all of the files, including the files that are staged out, in the /data1 filesystem on a system named “emc” to the /data2 filesystem on a system named “emc2.” This example will work on any system configured for HSM (fileserver and clients).

```
# ebcop -o /data1 | rsh emc2 /usr/epoch/bin/ebcop \ -i /data2
```

Note that **ebcop** does not update directory modification and access times, unless you issue the command a second time, using the **-dir** switch. See the **ebcop** man page for details.

Moving Files to Another EDM (Media Included)

The following example copies all of the files that are not staged out from the /data1 filesystem on a system named emc to the /data2 filesystem on a system named emc2. This example uses the **-local** switch so that **ebcop** reattaches files to the imported staging media rather than copy the staged-out files.

1. Insert all of the source machine’s staging media into the inlet of the destination machine’s library unit. *Do not* press any of the buttons on the front of the library unit.
2. Use the Library Unit Manager to import all of the volumes.
3. Run **ebfs_import -a** to complete the import.
emc2# /usr/epoch/bin/ebfs_import -a
4. Use **ebcop** to copy the files that aren’t staged out to the new system and to reattach staged-out files to their staged images. Note that this example only works with the primary staging media. The secondary staging media (the baseline backups) cannot be moved. (The **-R 1S** switch, which tells

ebcp not to copy the baseline information, is actually an option to **recxcpio**. See the **recxcpio** man page for details.)

```
emc# ebcp -o -local /data1 | rsh emc2 \ /usr/epoch/bin/ebcp -i /data2 -R 1S
```

5. After the copy has completed, you can delete the files on the source machine.

Moving Files Between HSM Clients (Store Included)

It is also possible to copy a large set of files, or an entire filesystem, from one network client to another. You can use **ebcp** to copy the magnetic-resident files to the new client and simply change the ownership of the client store from one system to another.

Note: If other filesystems are staging to the store, you need to repeat this procedure for each filesystem.

The procedure is as follows:

1. Bring the source filesystem to an inactive state.
2. On the EDM, use **emschs** to change the ownership of the client store from the source to the destination client.

The following command changes the ownership of the alpha_all store to a network client named beta:

```
emc# emschs alpha_all -c beta
```

3. On the EDM, use **emsmvs** to change the name of the store from alpha_all to beta_all:

```
emc# emsmvs alpha_all -n beta_all
```

4. If necessary, create a staging template on the destination system and configure the destination filesystem for staging.
5. Copy all the files from the data1 filesystem on the network client named alpha to the /data1 filesystem on the network client named beta:

```
alpha# ebcp -o -local /data1 | rsh beta \ /usr/epoch/bin/ebcp -i /data1 -R 1S
```

6. Repeat the procedure for all other client filesystems that stage to the same client store.

7. After the copy completes, delete the files on the source machine.

Restoring a Staged Out File That Has Been Deleted

When a file is staged out from an HSM client to a client store, the bitfile gets backed up by the fileserver's backup, and the file attributes that remain on the client get backed up by the client backup. If someone deletes the file, both the attributes on the client and the bitfile on the fileserver are deleted. You will not be able to access the file's data until the attributes have been restored on the client and the bitfile has been restored on the fileserver.

To restore the staged-out file:

1. On the client run **edmrestore** to start the EDM Restore window.
2. Select the client and work item.
3. Select the deleted file from the list.
4. Verify the destination and start the restore.
5. Run **emsundel** on the EDM server to restore the bitfile.

emc# **emsundel**

emsundel by default restores all deleted stores that have been recovered. The command should be run regularly from an entry in the root's crontab file. If you can wait, let the **emsundel** crontab entry restore the bitfile. The **emsundel** man page describes how restrict **emsundel** to a single store and force it to use a specific backup template.

Copying Files to a Non-EDM System

It is also possible to use **ebcp** to copy data from an HSM-enabled system to any Unix system. This example copies all files in the /data1 filesystem, including the files that have been staged out, on a system named "emc" to the /tmp/output filesystem on a system named "colt."

1. Change to the source directory:

```
emc# cd /data1
```

- 2. Mount the destination filesystem on your EDM:
emc# **mount colt:/tmp/output /mnt**
- 3. Use **ebcp** to copy the files:
emc# **ebcp . /mnt**
- 4. After the copy has completed, you can delete the files in the source filesystem.

Monitoring Storage Space and File Sizes

HSM provides several tools that enable you to monitor storage space and file sizes. These tools are listed in Table 13-2. See the man pages for further information.

Table 13-2

Monitoring Commands

If you want to:	Use this command:
Show magnetic disk space used by directories.	emdu
Show magnetic disk space used by directories and individual files.	emdu -a
Show virtual storage space used by staged-out directories and files; also shows magnetic disk space used by magnetic-resident directories and files.	emdu -av
Find out what staging volume a file resides on and show file sizes on magnetic disk and staging media.	emls
Show information about used and available disk space for each filesystem. (Syntax and display may vary from system to system.)	df
Show information about used and available inodes for each filesystem. (Syntax and display may vary from system to system.)	df -g

Table 13-2

Monitoring Commands (Continued)

If you want to:	Use this command:
List all the volumes in a staging trail and obtain information about current staging volumes.	dbreport appl_usage
Determine which staging volume to compact.	dbreport compaction
Display virtual filesystem statistics.	emfsreport

Maintaining Non-Stageable Filesystems

The root filesystem and the filesystem that contains the EDM software cannot be configured for HSM because they contain files that must not be staged out. These and any other non-stageable filesystems can therefore fill up.

The most likely reason that these filesystems would fill up is some sort of unexpected activity, either accidental or deliberate. If a filesystem becomes full, find files that can be deleted and determine why usage is increasing. Do the following to determine the cause of the problem:

- Use **ps** to determine what processes are running and kill any unexpected ones.
- Look at /tmp and /usr/tmp and delete any unnecessary temporary files.
- Use **find** to look for new files in the root filesystem.
- Examine the HSM log files in /usr/epoch/adm/archived.

Managing Your Magnetic Disks

If a filesystem is constantly staging files in and out, it may have an inadequate *working set*. The working set is the amount of stageable magnetic data that a filesystem can hold without exceeding its low watermark.

The working set is often measured as the number of days worth of files that are stored on the magnetic disk. This is called the *working set in days*. For most non-archival applications, you should have a working set period of at least one to four weeks. If your working set of files is considerably larger than your actual magnetic disk space, you will experience system performance degradation.

Use the **emfsreport** tool to display virtual filesystem statistics and to develop a coherent strategy for managing your magnetic disks:

1. Run **emfsreport** to display virtual filesystem statistics.
2. Decide how many days worth of data you need to keep on the magnetic portion of a filesystem.
3. Determine the additional magnetic disk space you'll need.
4. Reconfigure your magnetic disk usage or purchase more disks.

These steps are described in detail below.

Run emfsreport

Use **emfsreport** to find out your *working set size* and the *working set in days*, so that you can fine tune your system. To run **emfsreport**, become root and specify either the name of a locally mounted filesystem, or use the **-a** switch for all filesystems. For example:

```
emc# emfsreport -hva /data2
```

A sample report is shown in Figure 13-1. It displays the amount of space used by all files in the /data2 filesystem, regardless of their location. (Note that virtual space takes into account space consumed on magnetic disk plus space consumed on the staging media.)

Figure 13-1

emfsreport Output

/data2	TOTAL	REGULAR	DIR	SPECIAL	SYMLNK
Number of files	218502	105169	17160	0	96173
GB of stagable data	0.33680	0.33680	0.00000	0.00000	0.00000
GB of not stagable	0.11482	0.00511	0.01800	0.00000	0.09171
GB of data staged	1.20944	1.20944	0.00000	0.00000	0.00000
Virtual GB of data	1.59746	1.48775	0.01800	0.00000	0.09171
Stagable Vir-Phs ratio	4.743:1	4.417:1	1.000:1	1.000:1	1.000:1
Actual Vir-Phys ratio	3.537:1	4.351:1	1.000:1	1.000:1	1.000:1

GB available for working set: 0.353
(About 5.6 days worth.)

Histogram of virtual space by file age for regular files:

Range of days old	count	%	cum %	Kbytes	%	cum %	KB per
0 - 0.99	14489	13.78	13.78	158394	10.15	10.15	11
1 - 1.99	4746	4.51	18.29	58962	3.78	13.93	12
2 - 3.99	3778	3.59	21.88	69728	4.47	18.40	18
4 - 7.99	6572	6.25	28.13	106640	6.84	25.24	16
8 - 15.99	1738	1.65	29.78	39844	2.55	27.79	23
16 - 31.99	3679	3.50	33.28	88144	5.65	33.44	24
32 - 63.99	7687	7.31	40.59	221259	14.18	47.63	29
64 - 127.99	41762	39.71	80.30	488186	31.29	78.92	12
128 - 255.99	13370	12.71	93.01	162142	10.39	89.31	12
256 - 511.99	5789	5.50	98.52	122504	7.85	97.17	21
512 - 1023.99	1544	1.47	99.99	44013	2.82	99.99	29

Choose the Desired Working Set in Days

Using the values displayed in Figure 13-1, you can calculate how much more magnetic space you would need for a desired working set size in days. Figure 13-1 shows that /data2 has 353 MB available in its working set, with a working set in days of 5.6 days. Once you learn what your working set size in days is, you may decide that it is too small. (Remember, EMC recommends you have a working set period of at least one to four

weeks.) Select an adequate number of days based on your usage patterns. See “emfsreport and the Working Set” on page 11-31 and “Disk Utilization Zones” on page 11-13 for some additional considerations.

Determine Additional Magnetic Disk Space Required

The sample report of /data2 shows that the filesystem has a working set of 5.6 days. Suppose that is unacceptable and you'd rather have a working set of 14 days. How much additional magnetic disk space would you need?

To estimate, simply pick the high day range that falls closest to your desired working set size. In this case the day would be 15.99.

1. Multiply the value on the “Kbytes cum%” line (27.79) by the total number of Kbytes (1560023) to get the working set size in KB. For example:

Working set size in KB = $.2779 * 1,560,023$

The result is 433530.39 KB.

2. Take the result of the previous calculation (433530.39) and divide by 1024 to get the working set size in MB.

Working set size in MB = $433530.39/1024$

The result is 423.37 MB.

3. Take the result of the previous calculation (423.37) and divide by 1024 to get the working set size in GB.

Working set size in GB = $423.37/1024$

The result is 0.41 GB.

4. Subtract the GB available for the working set (0.353) from 0.410 and multiply by 1024 to get the number of additional magnetic space needed in MB.

Mag Disk Space needed = $(.410 - .353) * 1024$

The result is 58.37 MB.

Thus you would need approximately 58.37 Mbytes more magnetic disk space on this filesystem in order to have a working set of 15.99 days. (You can extrapolate to find the exact requirements for a working set size of 14 days.)

Reconfigure Magnetic Disk or Purchase More Disks

Depending on your site's configuration, your budget, and the amount of time you have, there are a number of actions you can take to better utilize your magnetic disk resources:

- Move files to another filesystem that is under utilized. This filesystem may be on the same or on another magnetic disk.
- Repartition your disk. Take a complete backup of your filesystems, repartition your disk and restore your files. When repartitioning your disk, make the partitions correspond to their working set needs; make some partitions smaller and some larger.
- Add magnetic disks if you are using all of your current disk space and have no where else to relocate the files.

Populating Filesystems

The most efficient way to move files from a non-HSM system to an HSM system is to make the HSM system an NFS client of the other system and to use **tar** to read the data. For example, to copy /user1 on another system to /data1/user1 on the HSM system, use the following procedure:

1. Configure the other system as an NFS server for its /user1 filesystem.
2. Login to the HSM system as root, make a temporary directory, and temporarily mount the server's /user1 as a remote filesystem:

```
emc# mkdir /other_user1
emc# mount -o,ro,hard,intr serv:/user1 /other_user1
```

3. Change to the remote filesystem and use **tar** in a pipeline to copy files to the new filesystem on the HSM system. Then, unmount the remote filesystem and delete the temporary directory:

```
emc# mount -F vxfs -o remount,nolog /data1
emc# cd /other_user1
emc# tar cBf - . | (cd /data1/user1; tar xBpf -)
emc# cd /
emc# umount /other_user1
emc# rmdir /other_user1
```

An HSM-enabled system can also be populated by configuring it as an NFS server, configuring the other system as the client, and using **tar** on the client system to “push” the files to the HSM system. Because NFS read performance is generally better than NFS write performance, the best approach is to make the HSM system the client and “pull” the files from the server.

When populating an HSM system with archival files, that is, files that you want to move quickly to staging media, you can set the *convenient* property (**-C**) on the directory that contains the files. Then, any file loaded into that directory (or subsequently created subdirectories) will be staged out during the next periodic staging run. See the **emchmod** man page for details about the convenient property.

For details on moving data from one EDM to another, see “Moving and Copying Files Between HSM Systems” on page 13-14.

Disabling Filesystem Staging

There are two ways to disable staging. You can temporarily disable periodic staging for an entire system, for a staging template, or for an individual filesystem. Or, you can permanently disable both periodic and demand staging.

Temporarily Disabling Periodic Staging

By changing the enable value from Y to N in either the **emsysconf** command, the **emstconf** command, or the **emfsconf** command, you can temporarily disable periodic staging for an entire system, for a staging template, or for an individual filesystem.

```
emc# emsysconf N 2OR
emc# emstconf CAD N - - - - - -OR
emc# emfsconf /mech N - - - - CADOR
```

See the man pages for details regarding command syntax.

The enable value only affects periodic staging; it has no effect on demand staging (crossing a high watermark), user-specified staging (**emstage** and **restage**), baseline backup, or stage-in.

Permanently Disabling Periodic and Demand Staging

Use the following procedure to permanently disable filesystem staging, that is, periodic and demand stage-out and stage-in. This procedure not only disables staging, but it also stages in all staged-out files. If the filesystem does not have the room, you can move the staged-out files to another filesystem. See “Copying Files from One Filesystem to Another” on page 13-14.

To preserve data integrity, you must follow this procedure exactly as described.

1. Make sure you have a valid set of backups for the filesystem.
2. Become superuser.
3. Change permissions on the root of the target filesystem to allow only *ruwx* access by root. (Before you change the permissions, do an **ls -lad** to determine what the current permissions are. You reinstate these permissions at a later date.)

```
emc# ls -lad /alpha
drwxr-xr-x 9 root 512 Sep 4 15:40 /alpha

emc# chmod 700 /alpha
```

4. Disable periodic and demand staging. Note that this step prevents periodic and demand staging, but it does *not* prevent users from staging specific files. At this point, users can still access staged out files.

```
emc# emfsconf -r /alpha
```

5. Disable compaction by commenting out or editing the appropriate line in root's crontab file. The default compaction line is as follows:

```
00 1 * * * /usr/epoch/bin/emcompact -c  
>/dev/null 2>&1
```

In addition, abort any compactions in progress.

6. Disable baseline backups from the Backup Configuration interface.
 - c. From the EDM main view's Backup menu, select Configure.
 - d. On the Server tab select **Disable baseline backups**.
 - e. Select Save, then exit EDM.
7. Shutdown the system and then reboot. This ensures that no one is logged in and severs any remote connections to the system.

Note: Use only `/usr/sbin/shutdown -y -i6 -g0` on an HSM system. Do not use `halt` or `reboot`.

8. Use **emfsreport** to determine whether enough free space is available to stage in all of the staged-out files. The important line to look at in the **emfsreport** display is the Virtual GB of data (239.35 MB, in this example):

```
emc# emfsreport /alpha
```

/alpha	TOTAL	REGULAR	DIR	SPECIAL	SYI
Number of files	16701	5799	1818	0	0
GB of stagable data	0.20034	0.20034	0.00000	0.00000	0.00000
GB of not stagable	0.01340	0.00288	0.00185	0.00000	0.00000
GB of data staged	0.03130	0.03130	0.00000	0.00000	0.00000
Virtual GB of data	0.23935	0.22884	0.00185	0.00000	0.00000
Stagable Vir-Phs ratio	1.195:1	1.142:1	0.000:1	0.000:1	0.000:1
Actual Vir-Phys ratio	1.120:1	1.126:1	1.000:1	0.000:1	1.000:1
GB available for working set: 0.230					
(About 572.1 to 739.6 days worth.)					
(Range given because 1% of virtual space					

9. If you do not have enough free space to stage in all of the staged-out files, refer to the following section.

10. Run **emfsdeconfig** to complete the procedure.

```
emc# emfsdeconfig /alpha
```

Moving Staged Out Files to Another Filesystem

If the filesystem does not have enough space to hold all of the staged-out files, proceed as follows:

1. Change to the /alpha directory.
2. Use **ebcp** to copy the files to another filesystem, which may be stageable or non-stageable. The following example copies all of the files in /alpha to /new_alpha.

```
emc# ebcp . /new_alpha
```

3. Delete the files in the /alpha filesystem.
4. Unmount the /alpha filesystem to clear data structures used for stageable filesystems.

```
emc# umount /alpha
```

5. Recreate the /alpha filesystem.

Compacting Staging Media

In most cases, staging media is compacted automatically via an **emcompact** entry in root's crontab file. With automatic compaction, **emcompact** automatically determines which volumes to compact.

You can also compact staging volumes manually, if you want to compact any additional volumes. Both automatic compaction and manual compaction use the **emcompact** command.

If you need to compact some staging volumes manually:

1. Use **dbreport**'s compaction report to decide which volumes to compact.

```
emc# dbreport compaction
```

The compaction report is divided into three sections. The most likely volumes to compact are those that are listed in the last few lines of the first section. These are the volumes with the highest percentage of stale files.

2. Use **emcompact** to compact the volumes. In the case of EOs, which have two sides, you need to specify each side, or volume, separately. You can specify volumes by:
 - volume id
 - sequence number (for single-sided media)
 - sequence number and side (for double-sided media)
 - barcode (for single-sided media)

The following example compacts both sides of disk #10:

```
emc# emcompact EO 10-1 10-2
```

You can type the command without any arguments to find out the legal media types. (In the case of tapes, you can specify a barcode.)

You can override HSM's file residence policy by running **emcompact** with the **-p** (policy) option. The **-p** option ensures that all files from the compaction source volume are staged out to the compaction output volume and none remain on magnetic disks.

Administering Compaction

There are several things you can do on a regular basis to make sure that compaction is working smoothly.

CAUTION: In order to ensure a complete file restore process, you should disable automatic compaction and **emvck** as soon as you realize that you've lost a filesystem or a significant portion of one.

- Check the Media Requests window every morning to see if automatic compaction has blocked.
- Keep a supply of blank, easily accessible and unlabeled volumes, which you can label and allocate as compaction output volumes.

To increase the likelihood of maintaining a supply of free volumes, you can also convert unrestricted staging trails to restricted ones, or prelabel volumes for a specific trail.

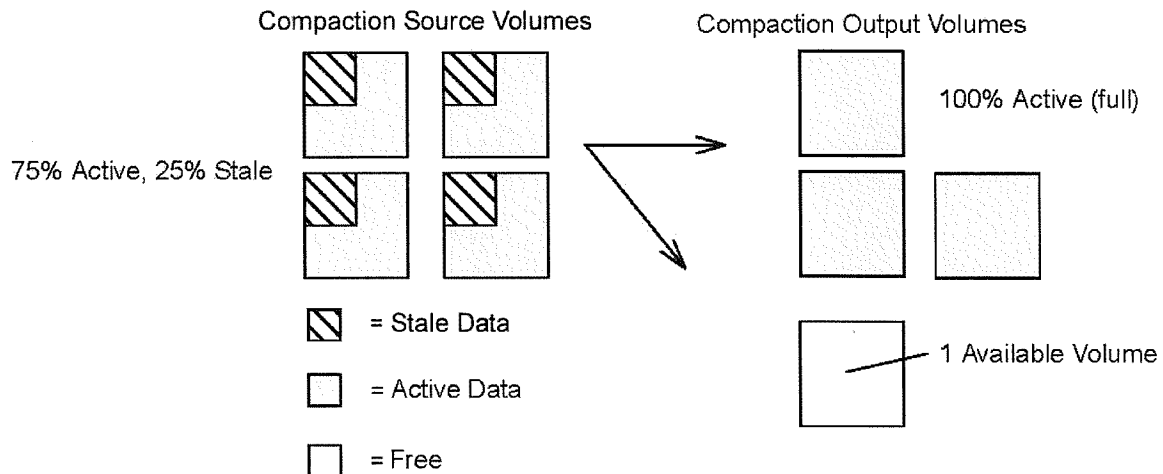
- Check the message files every day.

If automatic compaction frequently fails to reach the free goal, the system might have reached the limit of its data storage, given the number of volumes in the library unit or the time available for compaction.

In such a case, use the **dbreport appl_usage** report to determine whether there is enough stale space to reclaim, or whether you should add more volumes to your library unit. For example, in an EO system with volumes averaging 25% stale data, you must compact four volumes in order to get a single free volume (see Figure 13-2). If this rate is unacceptable in your environment, add more blank volumes. Refer to *EDM Server Error Messages* for further details on what to do if automatic compaction fails.

Figure 13-2

Freeing Volumes for Use



Compacting Baseline Media

Compacting baseline volumes is similar to compacting staging volumes, but you can only compact baseline volumes manually. Compacting baseline volumes causes all currently active data associated with the volume to be copied to the current compaction volume associated with the baseline trail. New baseline compaction volumes are allocated as necessary.

As a general goal, you should limit the number of active baseline volumes to the number of active staging volumes. If possible, you should also limit the number of active baseline volumes to the number of slots in your library units. This facilitates recovery in the event of a site disaster by ensuring that all of the baseline media (to which active files are attached) will fit in your library units.

To compact baseline volumes:

1. Run **dbreport baseline** weekly and refer to the "pct_stal" column for baseline volumes.
 emc# **dbreport baseline**

2. Use **emcompact** to compact the N most stale optical disks or tapes. (In the case of optical disks, there are two volumes per disk.) The value of N varies, but it is at least the number of active baseline media minus the number of active staging media or slots in your library units, whichever is less.

Therefore, if you have 55 active baseline media, 45 active staging media, and a 50-slot library unit, compact at minimum the ten most stale baseline media.

This results in a set of compacted baseline volumes. Note that these volumes are not deallocated and available for use until all existing baseline-relative backups that reference them expire.

Clearing Incomplete Bitfiles

As part of nightly maintenance, the system runs the **emscheck** program via root's crontab to clear the incomplete bitfiles that accumulated in the client stores as a result of an interruption of service during a stage-out.

For example, the following line in root's crontab runs **emscheck** every day at 12:30 a.m.:

```
30 0 * * * /usr/epoch/bin/emscheck >/dev/null 2>&1
```

Moving a Store to Another EDM Server

To move client stores from one EDM server to another, do the following:

1. Use the **emschs** command to freeze the client store.

```
emc# emschs alpha_all -z
```

Store Name

Freeze Store

2. Use the **ebcp** command to relocate the entire contents of the store. Note that you must copy both the magnetic and staged out data. The following example copies all of the

files, including the files that are staged out, in the /alpha_all store on a system named "emc" to the /alpha_new store on a system named "emc2."

```
emc# cd /stores/alpha_all
emc# ebc -o . | rsh emc2 /usr/epoch/bin/ebc
-i /stores/alpha_new
```

3. On the new server, emc2, use **emsmks** to add the store to the server configuration.

```
emc2# emsmks alpha_new -p /stores/alpha_new -c alpha_new
```

```
Directory /stores/alpha_new already exists, reuse [y/n]? y
Store configuration data already exists, reusing.
emsd is running, restarted..[OK]
```

4. Unfreeze the store on the new server:

```
emc2# emschs alpha_new -w
```

5. Notify the client system of the store's new location. Note that in this case trail_1 is the name of the previous staging trail.

```
alpha# emstconf trail_1 - - - - - emc2:alpha_new
```

6. Add the new migration tag on the new server:

```
emc2# emschs alpha_new -t new_migration_tag
```

7. Remove the old store from the original server:

```
emc# emsrms alpha_all
```

Gathering Migration Store Statistics

Use the **emsstat** command to display network migration server activity levels. The **emsstat** command gets its information by accessing statistics that are kept in a shared memory segment used by all active EDM Migration server daemon (emsd) processes, or from a statistics file if emsd is not running.

If you type **emsstat** without any arguments, it displays statistics representing server activity since the statistics were last cleared. If you use the **-i** option, **emsstat** displays the ongoing server activity, and, by default, updates the screen in 10 second intervals.

To reinitialize the statistics database enter the following commands:

```
emc# emshalt  
emc# rm /usr/epoch/etc/mal/emsd_stats  
emc# emsstart
```

See the **emsstat** man page for further details.

Checking a Network Client's Staging Configuration

Use the **emcheck** command to check the EDM Migration client configuration. You must be superuser to run **emcheck**. If you type the command without any arguments it checks the configuration database, warns you of potential problems and corrects inconsistencies. See "Checking the Staging Configuration" on page 13-12 for more information.

Troubleshooting HSM

HSM depends on the interaction of several daemon processes. Most HSM problems are caused by the failure of one of these processes. The following checklist enumerates the steps to take to troubleshoot HSM problems. (Use the **emlistd** command to check whether any of the following processes are running.)

- ☐ Verify that the HSM file monitor daemons (**emfmd**) are running.
- ☐ If you are unable to stage in files, verify that the stage-in daemons (**emsid**) are running.
- ☐ If demand or periodic staging fails, verify that at least one master daemon (**emmasterd**) is running.
- ☐ If the user-level commands (**emstage**, **embsi**, **emchmod**, and **emls**) are failing, verify that the HSM RPC daemon (**emrpcd**) is running.
- ☐ Look in the `/var/adm/epoch/detail` log for error message information.

Restoring a Lost or Damaged Staging Volume

The procedures for restoring a lost or damaged staging volume vary, depending on whether or not baseline backup is installed.

Restoring a Lost or Damaged Staging Volume (No Baseline Backup)

If baseline backup is *not* enabled:

1. Use the **dbreport volumes** report to find the volids of the missing volume(s). (The volid is a 16-digit hexadecimal number.) For two-sided media, you need to get the volids of both sides.
2. If the lost volume happens to be the current staging volume, the current compaction volume, or an active backup volume, you need to use **em_new_volume** as follows:

```
emc# em_new_volume staging_trail_name
```

3. Locate the files staged to the missing volumes by using **emfind(1)**:

```
emc# emfind / \( -staged_to 0000111122223333 -o\ -staged_to 0000111122223334 \) -print > /usr/tmp/files
```

Note: If **emfind** encounters a pathname that is too long, it generates an error message. The pathname is not added to /usr/tmp/files, and the subsequent restore are incomplete. Some manual intervention (descending into directories and rerunning **emfind**) is necessary in such cases.

4. Delete the missing volumes from the database using **evrmvol**:

```
emc# /usr/epoch/bin/evrmvol -v 0000111122223333
```

```
emc# /usr/epoch/bin/evrmvol -v 0000111122223334
```

5. Restore necessary files:

```
emc# ebrestore -D server -c server -w workitem -d -f /usr/tmp/files
```

Restoring a Lost or Damaged Staging Volume (Baseline Backup is Enabled)

If baseline backup is enabled, you can use **ebcheck** to restore a staging volume. The steps are as follows:

1. Using **dbreport volumes**, obtain the IDs of the missing volume(s). For two-sided media, you need to get the IDs of both sides.
2. Delete the missing volumes from the database using **evmrmvol**:

```
emc# /usr/epoch/bin/evmrmvol -v 0000111122223333
```

```
emc# /usr/epoch/bin/evmrmvol -v 0000111122223334
```

3. If the missing volumes are baseline backup volumes, simply invalidate the pointers to the baseline volumes:

```
emc# ebcheck -a -i
```

4. Otherwise, if the missing volumes are primary staging trail volumes, restore as many of the files as possible from the baselines:

```
emc# ebcheck -a -i -b1 -b2
```

5. Restore any remaining files from the full and incremental backup volumes:

```
emc# ebcheck -a -r1 -r2 > /usr/tmp/files
```

6. If /usr/tmp/files is non-empty:

```
emc# ebrestore -D server -c server -w workitem -d -f /usr/tmp/files
```

Restoring a Lost or Damaged Staging Trail

This method uses baseline volumes as a temporary staging trail, which lets you bring the system back up much sooner. The procedure is very quick, and you can generate your new primary staging trail while the system is up and providing service:

1. Delete each missing volume from the database as before.
2. Make the baseline backup volumes also act as a copy of the primary staging trail:

```
emc# ebcheck -a -i -c1 -c2
```

3. Restore any remaining files from the full and incremental backup volumes:

```
emc# ebcheck -a -r1 -r2 > /usr/tmp/files
```

4. If /usr/tmp/files is non-empty:

```
emc# ebrestore
```

5. Using **restage**, generate a new primary staging trail for each filesystem as appropriate:

```
emc# restage -t trailname / \ ( -fstype nfs -prune \) \ -o -staged_to_trail  
baseline_trailname
```

Restoring a Lost or Damaged Filesystem

These steps restore a filesystem to its state as of its last backup, after a complete loss due to disk failure and the like. The steps assume the staging trails, backup catalogs, and the root filesystem all are still intact. The steps are designed to avoid all accesses to the filesystem while it is being restored, whether local or remote, so that users do not see a filesystem that is only partially recovered.

1. Create a new empty filesystem, the same size as the original one or larger:

```
# mkfs -F vxfs -o inosize=512 /dev/rdisk/cXtYdZ
```

2. Mount the filesystem to a temporary location (called /tmp_doc in the examples below):

```
# mkdir /tmp_doc
```

```
# mount -F vxfs /dev/cXtYdZ /tmp_doc
```

3. If the filesystem was configured for HSM, you must now configure the new filesystem for HSM. This step must be done before recovering any data, or staged out files are not recovered correctly.

Use the EDM HSM configuration interface, just as if you were configuring a filesystem for the first time. The filesystem may be assigned to the same staging trail as the original, or to a different staging trail; it makes no difference.

4. Using the EDM Restore interface, restore the last backup of the filesystem you lost (called doc here) onto the new filesystem (mounted here at /tmp_doc).

Mark everything in the filesystem except for the `.-EPOCH-` and `lost+found` directories.

5. Change to the `/tmp_doc/doc` directory, and move everything to `/tmp_doc`.

```
# cd /tmp_doc/doc
# mv * /tmp_doc
# cd ..
# rmdir doc
```

Be sure to check for files or directories whose names begin with `.-` since they are not moved by the `mv` command. The **rmdir** command fails with "Directory not empty" in such cases. The **ls -a** command lists these files; they have to be moved manually.

6. If the new filesystem (`/tmp_doc`) was configured for HSM, remove it now from the staging configuration database. There is no need for this configuration now, since the original filesystem (`/doc`) should still be in the staging configuration database.
7. Umount the filesystem from its temporary mount point, and remount it under its real name.

This example assumes that the same physical device address is being used for the new `/doc` (the usual case if a new disk was brought in to replace a failed one). If the device address is now different, be sure to update the `/etc/vfstab` entry for the filesystem.

```
# cd /
# umount /tmp_doc
# rmdir /tmp_doc
# mount /doc
```

Part III Logs and Reports

14 **Start of Backup and Related Processing**

The EDM Backup software automatically initiates backups, processes catalogs, and generates backup reports. You can also initiate these functions manually, or edit the crontab file to change how these functions occur automatically. This chapter contains the following topics:

- Backup Processing
- Catalog Processing

Backup Processing

Following are methods of initiating backup processing with EDM Backup:

- by using the Backup Activity Wizard
- automatically from crontab (see page 14-3)
- manually from command line (see page 14-7)

The Backup Activity Wizard enables you to start new, queued, or failed backups, stop running backups, or manage the backup queue. You access this wizard from the Main window of the EDM GUI.

Note: You must have root privileges or be an EDM Backup Administrator to use the Backup Activity Wizard.

For automatic processing, EDM Backup uses the crontab facility to issue the **ebbackup** command at a particular time each night to start backups. The command specifies a backup schedule template, which contains scheduling parameters. You can create additional crontab entries (using the Backup Configuration Wizard, by editing the file, or from the Backup Configuration window of the EDM GUI) for any schedule templates that you add. You can optionally specify a particular work group or work item, backup level, or specific day with the **ebbackup** command, but you would typically use the Backup Configuration Wizard or the backup template.

To start network backups manually, you can issue the **ebbackup** command from the command line. Again, you can choose to specify a backup schedule template, but you would more typically specify a particular work group or work item, backup level, or specific day when starting backups from the command line.

To start Symmetrix Connect backups manually after successful client configuration, issue the **eb_dc_backup** *work_item* *_base_name* command. Refer to the *EMC Data Manager Symmetrix Connect User Guide* for more information on starting these types of backups and related processes.

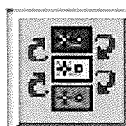
Note: However you initiate backups, make sure your clients are configured and that you have labeled your media and inserted that media into the appropriate library unit.

For additional information, refer to “Scheduling” on page 3-4.

The following sections describe starting backups in the crontab file or manually from the command line.

Backup Activity Wizard

As mentioned above, the Backup Activity Wizard enables you to start new, queued, or failed backups, stop running backups, or manage the backup queue. You access this wizard from the Main window of the EDM GUI.



Click this icon in the Main Window to start the Backup Activity Wizard.

In the wizard panels, you select a backup operation, select the objects on which you want to operate, choose backup options, and confirm your actions. You can then monitor the progress of the backup operation that you initiated in the Main window.

Refer to EDM online help for more information.

Automatic Nightly Processing

You can use one of the following methods to configure the crontab facility to schedule automatic backups for each defined schedule template in the configuration file:

- Backup Configuration Wizard (which you access from the EDM Main window toolbar)

- By default, the backup schedule template runs automatically at 6:00 p.m. every night.

Each line in root's crontab file has several fields of information. Figure 14-1 shows the format of the crontab entry that invokes the **ebbackup** program.

Figure 14-1

Root's Crontab File Entries

Minute

Hour

Day of Month

Month of Year

Days of Week

Command to Run

Argument(s)

You can also schedule a backup in the crontab file within the Backup Configuration window of the EDM GUI. In the window, you select a work group for backup and the time that the backup is to occur. You can also indicate whether you want to schedule a failed backup, or use new media for a backup. (For more information, refer to Chapter 2 of this manual, EDM online help, and the **ebbbackup** man page.)

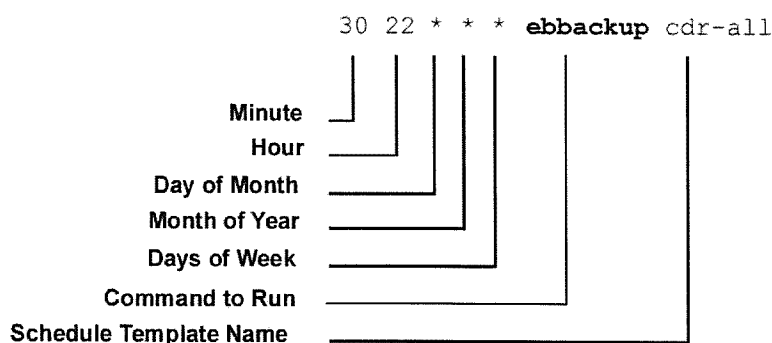
After you edit the file, the backups occur automatically. At the specified time each day, root's crontab file invokes the **ebbackup** command, which starts overall backup processing. Edit the file again only when you want to change the nightly backup run time.

Figure 14-2 shows a sample crontab entry, which starts a backup using the schedule template named "cdr-all."

The template "cdr-all" specifies the work groups to back up and the trailset to write the backups to. Asterisks(*) represent unspecified fields.

Figure 14-2

A Sample Crontab File Entry



The sample crontab entry in Figure 14-2 indicates the following:

- the Minute field specifies to start the backup 30 minutes after the hour.
- the Hour field specifies the hour at which to start the backup. In this example, the backup will start at 10:30 p.m. The length of time **ebbackup** runs depends on the length of the shift as defined in the backup template or when all scheduled backups finish—whichever occurs first.
- the Day of Month field indicates all days of the month (*).
- the Month of Year field indicates all months (*).
- the Days of Week field indicates all days of the week (*).

- the Command to Run field specifies the command (**ebbackup**).
- the Schedule Template Name field specifies the backup schedule template on which to run the backup (cdr-all).

Automatic Scheduling

The **ebbackup** command starts backups for a backup schedule template. By default, a backup schedule template specifies automatic scheduling of backups for all of the work items affected by it. When EDM Backup starts it uses the rotation period, the rotation scheme, and the backup shifts that are specified in the named schedule template to determine what work items to back up during that backup session.

EDM Backup manages the backups, performing one full backup (level 0) for each work item each rotation period, and scheduling level 9 backups on the remaining days in the rotation period.

EDM Backup's ability to automatically balance the backup work load frees you from the task of manually assigning each client to a backup schedule. Automatic scheduling also adjusts to clients that are down during the scheduled backup.

Custom Scheduling

You can also specify custom scheduling for the backup. In custom scheduling, the backup schedule template explicitly defines the work items and the days on which they are scheduled for backup. You can specify the custom schedule for the template within the Backup Configuration Wizard or the Backup Configuration window of the EDM GUI.

Backups that you schedule by using the custom schedule directives in the backup template are initiated using root's crontab file to begin processing for the template, just as for automatic scheduling.

Note: If a client is unavailable on its scheduled backup day, the backup does not automatically reschedule the backup to another day as auto scheduling does.

It is also possible to use the **ebbackup** command to directly specify particular backup levels on certain days for certain work groups or work items. This alternative command format is described in the next section.

Command Line Processing

You can choose to initiate all backups by using the **cron** facility or command line.

The **ebbackup** command enables you to specify the level of backup, work group, work item, priority, and/or trailset to use for a backup. You can use the **ebbackup** switches in combination with the crontab facility to schedule a command for a certain time each day to schedule each backup.

The **ebbackup** command options are useful when you want to schedule a specific backup. For example, you may want to force an immediate full (level 0) backup for a particular client's work item before performing an operating system upgrade on that client. You can also use the command line to schedule or reschedule backups that have failed. Refer to the man pages for **ebbackup** for details about the available options.

When you specify a backup from the command line, this backup overrules any backups scheduled by automatic or custom scheduling.

Note: To run a command line backup for a particular backup level, you must first define that level in a trailset.

The following examples show how to use the command line to backup a work group, a work item, and an HSM work item. For more information see the **ebbackup** man page.

The command line in Figure 14-3 starts level 9 backups for the clients in the work group named "cad."

Figure 14-3

Backing Up a Work Group

	#	ebbackup	-l	9	-g	cad	cdr
ebbackup Command		_____					
Backup Level Switch			_____				
Backup Level				_____			
Work Group Switch					_____		
Work Group Name						_____	
Backup Schedule Template Name							_____

The command line in Figure 14-4 starts level 0 backups for the work item named “cad-all.”

Figure 14-4

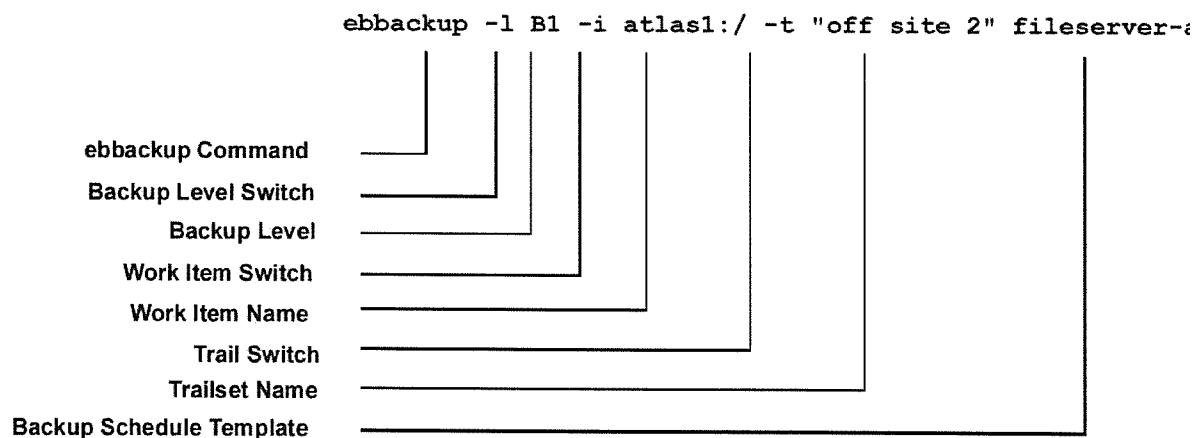
Backup up a Work Item

	#	ebbackup	-l	0	-i	cad-all	cdr-all
ebbackup Command		_____					
Backup Level Switch			_____				
Backup Level				_____			
Work Item Switch					_____		
Work Item Name						_____	
Backup Schedule Template Name							_____

The command line in Figure 14-5 starts baseline backups for the HSM work item “atlas1:/” to the “off site 2” trailset.

Figure 14-5

Backing Up an HSM Work Item to a Trailset



Catalog Processing

When a backup completes, the raw data for the associated catalog exists on the EDM Backup server. The catalog daemon (**ebcatalogd**) must process the raw catalog before the restore process in the EDM Restore window or the command **ebrestore** can use it. You can have catalog processing performed concurrently with backups, or you can schedule catalog processing for a later time so that this task does not slow down client backups.

Normally, the `/etc/rc3.d/s30ebs` script starts **ebcatalogd** at boot time.

To exercise greater control over the catalog processing schedule, you can start and stop **ebcatalogd** manually, or you can control it automatically via root's crontab file. To start catalog processing, run **ebcatalogd** without arguments. The catalog daemon places itself in the background when it runs, and terminates if another catalog daemon is already running. To stop catalog processing, run **ebcatalogd** with the **-halt** option.

Figure 14-6 shows a crontab entry that starts catalog processing at 6:30 AM each day, which is after the site's backup shift ends.

Figure 14-6

Starting Catalog Processing from Crontab

```
30 06 * * * /usr/epoch/EB/config/daemon_startup -ebcatalogd
```

Figure 14-7 shows a crontab entry that stops catalog processing an hour before the backup shift begins again at 10:30 p.m., the site specifies halting catalog processing at 9:30 p.m.

Figure 14-7

Halting Catalog Processing from Crontab

```
30 21 * * * ebcatalogd -halt
```

15 Message Logging

The EDM Backup and HSM software maintains a message logging system that uses both the system log daemon, **syslogd**, and circular logs to record significant events. These messages can be written to log files or to the system console. The configuration file (`/usr/syslog.conf`) determines the error conditions that are logged and where the messages are sent.

This chapter contains the following sections:

- Message Logging Features
- Syslog Message Files
- Circular Log Files
- Log Message Format
- Default syslog Configuration File

Message Logging Features

The message logging system offers the following features:

- Automatically creates log files:

During system installation several message log files are created automatically. These files are specified in the `/etc/syslog.conf` file, and described in “Default syslog Configuration File” on page 15-6.

- Uses **cron** to mail messages to system administrators:

For sites with a mail facility, **cron** starts a script that mails log messages which describe the system activities (anomalies only) for the previous 24-hour period to the system administrator. (See “Running Procedures Automatically via Cron” on page 2-6.)

- Groups files into logical categories:

The **syslogd** sends log messages to specific log files depending on the type of message. Because EDM Backup software groups messages into separate log files, you can choose to look at the log file that is most appropriate for the task at hand. For example, all messages that describe maintenance activities are sent to one particular log file, while messages that show error audit trails are sent to another.

- Monitors interaction of EDM subsystems:

Subsystem messages work together to give a complete view of system activity. This means that when an event occurs, all affected subsystems log a message. By looking at messages that are sent from different subsystems, you can see the relationship of each subsystem's activity.

Syslog Message Files

All syslog messages are written to log files that reside in `/var/adm/epoch`.

Note: All of the log files, except for the daily log, are rotated or archived in `/usr/epoch/adm`, where they remain for four weeks.

The messages in these files enable you to determine the cause of a system problem. The log files are named the following:

- concise
- daily
- debug
- detail
- mntfault
- lu_hardware

Table 15-1 explains when to look at each message file:

Table 15-1

When to Look at the Syslog Message Files

If you want to:	Look at this file:
determine quickly if any system problems have occurred. If the file is empty, you know the system is operating properly. You should view this file daily.	<code>/var/adm/epoch/concise</code>
see system problems for the past 24 hours. The daily log is a subset of the concise log.	<code>/var/adm/epoch/daily</code>
check the debug log. Debugging must be turned on according to the directions in the <code>/etc/syslog.conf</code> file. This file is primarily for use by Customer Service personnel.	<code>/var/adm/epoch/debug</code>
see additional information about the errors that appear in the concise log.	<code>/var/adm/epoch/detail</code>
view a list of volume requests (which require operator assistance) and the audit trail for volume allocation and erasure. You can forward these messages to other systems and/or to the system console for monitoring.	<code>/var/adm/epoch/mntfault</code>
see additional information about hardware errors that appear in the detail log	<code>/var/adm/epoch/lu_hardware</code>

Circular Log Files

You can configure message logging to bypass the standard system logs and write messages to circular logs. Circular logs are located in a central directory or in application-specific directories. The following directories contain circular logs:

- `/usr/epoch/adm/circular`
- `/usr/epoch/etc/subdirectory`

The names of the circular log files that are located in the central directory `/usr/epoch/adm/circular` are based on the daemon name. For example, the name of the circular log for the volume management erase daemon is `vmemd.log`.

Volume management maintains its circular log in `/usr/epoch/etc/vm`. This provides the system administrator with access to all VM-related files in one directory. For this same reason, circular logs for each Library Manager reside in individual subdirectories of `/usr/epoch/etc/lm`.

Circular logs that reside in VM and LM directories are named `clog` (for circular log).

You can use the **fuser** command to show all processes that have a given file open. For example:

```
# fuser -f /usr/epoch/etc/lm/hp_mf_sa/clog
```

Log File Rotation and Archival

If you are using the migration option, the concise and mount fault logs are rotated in `/usr/epoch/adm`.

The detail and debug logs are archived in `/usr/epoch/adm` for one year in systems on which EDM Migration is installed. The logs are archived in *year-month* subdirectories as shown in the following examples:

```
/usr/epoch/adm/1999-10
```

```
/usr/epoch/adm/1999-11
```

```
/usr/epoch/adm/1999-12
```

Within each subdirectory, the archived logs are written to a *detail.day* file. For example, the following file contains the detail log for December 23, 1999.

```
/usr/epoch/adm/1999-12/detail.23
```

If more than one rotation occurs on a single day, another suffix is added as shown in the following examples:

```
/usr/epoch/adm/1999-12/detail.23.0
```

```
/usr/epoch/adm/1999-12/detail.23.1
```

```
/usr/epoch/adm/1999-12/detail.23.2
```

The highest number suffix (detail.23.2 in this example) represents the most recent log.

Log Message Format

Each log message provides the following information:

- date/time string
- host name that identifies the name of the system on which the event occurred
- name that identifies the process that generated the message.
- optional user ID (usually supplied if an interactive program is logging the message). Many subsystems can only be run by root and therefore omit this field.
- process ID number that appears between square brackets. (Kernel messages, which are identified by the prefix *vmunix*, do not have this field.)
- optional layer name, sometimes prefixed by a subsystem name, that provides EMC customer service with additional information
- message number that uniquely identifies the message. Using the prefix that immediately precedes the pound sign (#). This prefix is either the process or layer name, depending on the message.

- brief free-form description of the condition.








To summarize, each log message conforms to the following syntax:

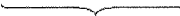

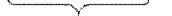

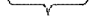


```
date/time host process <as user> pid <<subsystem:> layer:> message # -- message
```

Figure 15-1 lists some sample log messages:

Figure 15-1

Sample Log Message

Dec 11 03:11:18	pooh	backup	by root	[505]	#456	- - Backup starting
						
date/time string	host	process	user	pid	message #	message

Jan 28 11:31:07	pooh	VM as root	[33]	ELM:VM:	#101	- - VM Starting up
						
date/time string	host	process	pid	subsys:layer	message #	message

Default syslog Configuration File

A default, sample /etc/syslog.conf file is provided when you install the software. It contains the following lines in which the first column specifies the error condition and the second column specifies the file to which the error is logged.

```
kern.err;local5.err    /var/adm/epoch/concise
kern.err;local5.err    /var/adm/epoch/daily
local5.warning         /var/adm/epoch/mntfault
kern.info;local5.info  /var/adm/epoch/detail
*.debug               /var/adm/epoch/debug
```

Note: The concise and daily logs receive the same messages. The messages in the daily log are truncated each day.

16 Backup Reports and Log Files

When you run a backup or restore, EDM Backup saves information that you can then access through log files or reports. You can review this information to verify that your applications are executing correctly.

Reports are available online in the Backup Report window of the EDM GUI. You can execute reports on successful, failed, active, and queued backups on your local EDM and on multiple EDMs which are set up in a domain. This window enables you to create, modify, and print backup reports to look at key areas, such as performance within specified time periods, work items with poor performance, or failed work items. (For more information see the online help for the EDM Backup Report window.)

Some reports are generated automatically. You can run the **ebreport** commands for other reports from the command line or insert them into your crontab file for automatic processing.

This chapter describes the following reports and logs (for more information, refer to the appropriate man page).

- Report and Log Usage
- Executing Reports from the EDM GUI
- Report and Log Summaries
- Backup Reports
- Backup Media Reports
- Backup Duplicate Reports
- Backup History Reports
- Backup Disaster Reports
- Backup Baseline Reports
- Backup Completion Reports
- Backup Failure Reports
- Backup Coverage Reports
- Volume Reports
- Log Files

Report and Log Usage

You receive logs automatically, whereas you need to request most reports. After a backup or restore finishes, you receive an e-mail message that indicates whether the operation was successful. (You can also monitor backups in progress and then generate reports via the EDM graphical user interface (GUI); refer to “Executing Reports from the EDM GUI” on page 16-3).

If the backup or restore is successful, you do not have to review any of the logs and reports. However, these reports include information about your system that you may want to know or monitor.

If your backup fails, you receive an email message that tells you the backup failed. You should then look at the backups.log file, which contains clues as to why the backup failed. If you see a message such as “Client not available,” a client may have gone down during the backup or your system may have a network problem. You should review the backups.log file on the client that failed to see what information that file has on the backup.

A MINIMAL disaster report is generated automatically each time the LOCAL_DATABASE work item completes. You need the information in this report to perform a disaster recovery. The disaster report, by default, is mailed to the backup administrators, printed, and saved to disk.

Note: Refer to Chapter 19 “Being Prepared for a System Disaster”.

Regardless of whether your backup succeeded or failed, you may want to review the backup coverage reports. Each report displays information about the filesystems that were not backed up. It displays the client names, the size of each filesystem, the name of each filesystem, and a summary line with grand totals for each column of information.

Executing Reports from the EDM GUI

You can execute reports in the EDM graphical user interface (GUI). These reports can be reports for a local EDM or domain reports for a group of EDMs. The online help for the Backup Report window explains how to set up and use a domain, and lists the limitations of domain reporting (for example, that a domain cannot span a firewall or reconcile time differences on machines).

Objects in the Main window such as the EDM server, a client, or a work item, are colored to designate a successful, failed, or queued backup.

You can configure, run, and print backup reports on specific areas such as failed work items or work items with poor performance.



Click on this icon in the Main window toolbar to access the backup report module.

For more information about active backup reporting, refer to EDM online help, “Backup Report Overview.”

Report and Log Summaries

You can initiate reports manually by using the following commands, or you can add them to your crontab file for automatic generation. For more information on a specific report, refer to the man page for the report.

Table 16-1 lists the reports that EDM Backup generates.

Table 16-1

EDM Backup Reports

Report Name	Command	Information
Backup	ebreport backup	Contains a summary of the backup activity performed by the server.
Backup media	ebreport media	Contains information about the media to which EDM Backup wrote backup data.
Backup duplicate	ebreport duplicate	Contains information about media duplication processes. (Refer to Chapter 9 "Media Duplication".)
Backup history	ebreport history	Displays information about the backups performed on the server. Use the command options to display specific details.
Backup disaster	ebreport disaster	Contains a combination of other reports including a list of media for each backup trail, a detailed record of client backups, copies of the key configuration-file settings, and a history.
Backup baseline	ebreport baseline	On HSM systems, contains a summary of baseline backup activity as reflected in the saveset database.
Backup completion	none ¹	Confirms backup operations.
Backup failure	none 1	Notifies you if your backup fails.
Backup coverage	ebreport coverage	Displays information about which filesystems were backed up and which filesystems EDM Backup is not backing up.

1. There is no command available since the report is generated automatically.

For detailed descriptions, see the following sections in this chapter.

Note: The examples in this chapter are from different servers and different configurations and times. They cannot be compared to each other.

Table 16-2 lists the log files that are generated by EDM Backup and located in `/usr/epoch/EB/log` on the server. For detailed descriptions, see “Log Files” on page 16-35.

Table 16-2	EDM Backup Log Files
Report Name	Information
backups.log file	Displays an audit trail of backup-related activities on the server and the client.
recoveries.log file	Displays ebrestore operations on the server and the client.
ebcat.log file	Contains catalog processing startup and shutdown times.
Server log file (template level)	Displays backup information about activities for a single backup schedule (template).

Backup Reports

The **ebreport backup** command presents a summary of EDM Backup activity. It reports the status of the *most recent* backup run (unless the **-recent** or **-since** option is given). The work items are grouped by template (backup schedule) name, and trailset (media set). For a given work item name, the most recent backup is shown first.

Run **ebreport backup** every day to verify that all of the scheduled backups completed.

Table 16-3

ebreport backup Command Information

Option	Argument Definition	Description
-client <i>clientname</i>	<i>clientname</i> is the client whose backup history you want to display.	Displays only the specified client's work items that are backed up to the backup server. When the client name used is <i>servername</i> , it displays backups of server filesystems and databases, but does not display backups of filesystems or databases on remote clients.
-level <i>levelnumber</i>	<i>levelnumber</i> is a level from 0 to 9, a range of levels (for example, 0-8), or B1 or B2, for which you want to display backup history.	Displays only the backups of the specified levels (unless you use the other options to limit the report coverage).
-recent		Displays all of the work items that were backed up by EDM Backup, from the second most recent level 0 backup to the present.
-since <i>date</i> [<i>time</i>]	<i>date</i> is the date in the format <i>mm/dd/yy</i> . <i>time</i> (optional) is the time in the format [<i>hh:mm[:ss]</i>].	Limits the backup report to a range of dates. Use -since to show the backups that occurred on or after a particular date or use -until to display the backup history that occurred on or up to a particular date.
-until <i>date</i>		

Table 16-3

ebreport backup Command Information (Continued)

Option	Argument Definition	Description
-trailset <i>primary</i> <i>alternate</i>		Displays only the work items that were backed up by a primary or alternate set of media (trailset).
-template <i>name</i>	<i>name</i> is the template (schedule) from which backups are selected. If "*" is used for <i>name</i> all templates are selected (the default). Note that "*" must be quoted on the command line	Selects only the backups that were created for the named template.
-workitem <i>name</i> or -item <i>name</i>	<i>name</i> is the named work item for which backups were created. If "*" is used for <i>name</i> all work items are selected (the default). Note that "*" must be quoted on the command line.	Selects only the backups that were created for the named work item.

You might see database work items with the same name except for an added suffix in the form ":stripe_n_of_m." This occurs if the backups were *striped*. If the backups were striped, you also see a suffix on the template names. For example:

```
hamster:master+:sybase:stripe_1_of_2.
```

Figure 16-1 shows a sample backup report.

Figure 16-1

EDM Backup Backup Report

EDM Backup Backup Report for server tesla at Sept 28 09:09:09 1998
Report options: -client tesla
Template name, Primary/Alternate: Trailset name
====

ebfs_bt_1, Primary: ebfs_ts_1									
Work item name	Level	Start time	Time used	Backup	Files\	bad	Size	Catalog	
ebfs_wi_1	0	10/04/98 06:43:32	0:12:30	Completed	33\	0	430.0 MB	Complete	
ebfs_wi_2	0	10/04/98 06:43:32	0:11:40	Completed	37\	0	401.0 MB	Complete	
ebfs_wi_3	0	10/04/98 06:43:32	0:12:10	Completed	46\	0	415.0 MB	Complete	
ebfs_wi_4	0	10/04/98 06:43:32	0:11:30	Completed	29\	0	393.0 MB	Complete	

Backup Level

Backup States

Catalog States

Table 16-4 describes the different backup states that can appear in the Backup Report.

Table 16-4

Backup States

State	Description
Started	Backup is underway (or it was interrupted before finishing)
Partial	Backup was manually shut down before finishing
Incomplete	An error caused the backup to fail
Completed	Backup finished successfully
Unsuccessful	Backup completed with errors
Failed	No files were backed up, possibly due to a misconfigured work item
Timed-out	Client connection timed-out

Table 16-5 describes the different catalog states that can appear in the Backup Report.

Table 16-5	Status of Catalog Processing
State	Description
Partial	Catalog is being created (or it was interrupted before finishing)
Unsorted	Intermediate state in the post processing of the catalog.
Sorted	Intermediate state in the post processing of the catalog.
Complete	Post processing completed
Delta	Catalog was reduced to a collection of changes against the catalog of the subsequent backup
Expired	Online catalog for the backup was deleted

Backup Media Reports

You can use the **ebreport media** command to display a list of all volumes to which EDM Backup wrote backup data. Use one of the options in Table 16-6 to limit the report size.

Table 16-6

ebreport media Command Information

Option	Description
-active	Displays the volumes that EDM Backup is currently using to write backup data.
-offsite	Displays the volumes that are marked for offsite storage.
-onsite	Displays the volumes that are marked as onsite (usually after their status was changed from offsite to onsite).
-template <i>name</i>	Displays the volumes that the named template uses.
-trailset <i>name</i>	Displays the volumes that the named media set (trailset) uses. If -template is used, only "Primary" or "Alternate" (both of which are defined in the Backup Configuration window), can be specified.
-trail <i>name</i>	Displays the volumes that the named trail uses within a trailset.
-orphans	Displays a list of orphaned volumes. This cannot be combined with any other command line option.
-no_baselines	Prevents the baseline media report from being generated on an EDM with the HSM option. This report is never displayed on a system without HSM.
-help	Displays command line options for ebreport media .

A *media rotation* is the collection of volumes that a single backup schedule template writes to a particular trailset (media set) and trail during a single scheduled rotation period.

Volumes are grouped into media rotations. The media report contains one section for each schedule template, trail, and trailset. The report lists within each section the volumes for each media rotation with the date, rotation ID, number of backups, and whether media duplication was used.

Figure 16-2 shows a sample report that **ebreport media** generates:

Figure 16-2

EDM Backup Media Report

Summary of all media, listed by media rotation groups

Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary Trailset

09/30/1998 12:54:42 Rotation ID:4CD84987.F6BECF8D.00000200.54028F30, 4 backups

Media duplication used on 1 copy

*Orig Vol: 60D84A1170094B3E (BNY574), Seq #: 000024 in TLU: at_dlt_3264_0, media: DLT tape

Dup Vol: 73D8745B3E0384A5 (BDE133), Seq #: 000028 in TLU: at_dlt_3264_0, media: DLT tape

Duplication State: Done, Successful, Duplication Date 05/08/1999 16:06:04

Descriptions

Section Header

Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary
Shows Template Name, Trail Name, and Trailset.

Rotation Header

09/30/1998 12:54:42 Rotation ID:4CD84987.F6BECF8D.00000200.54028F30, 4 backups

Media duplication used on 1 copy

Shows Backup Date and Time, Rotation ID, Number of Backups on Media, Use of Media Duplication.

Volume Entries

*Orig Vol: 60D84A1170094B3E (BNY574), Seq #: 000024 in TLU: at_dlt_3264_0, media: DLT tape

Dup Vol: 73D8745B3E0384A5 (BDE133), Seq #: 000028 in TLU: at_dlt_3264_0, media: DLT tape

Shows Asterisk for most recent volume in the rotation, Blank for all others.

Then Original or Duplicate, Volume ID, (Barcode), Volume Sequence Number, TLU Type, and Media Type.

Duplication information follows with Duplicate Volume ID, (Barcode), Volume Sequence Number, TLU Type, Media Type.

The Duplication State, Duplication Date and Time follow.

Backup Duplicate Reports

The **ebreport duplicate** command displays a list of the original and duplicate volumes that are currently allocated to Epoch-Backup. These volumes are grouped into media rotations. Media rotations in the report are grouped by template, trailset, and trail.

The first line in the report provides the status for the entire rotation; one or more lines follow that show the volumes allocated to the rotation. The rotation status line contains the time that the rotation was created, the rotation ID, the number of partial or complete backups written to the rotation, whether media duplication is enabled for this rotation, and if so, how many duplicates were made. For each original volume the following duplication information appears: duplication state of the volume (Scheduled, Done, etc.), whether the duplication is up to date, duplication status (Empty or Old), and mode of duplication.

Information for each original volume in the report includes the 16-digit volume ID, volume barcode in parentheses, volume sequence number, library unit in which the volume resides, and media type (e.g., DLT or EO). A "*" precedes the last original volume that was allocated to the rotation.

If the original volume has an allocated duplicate volume, the line for each duplicate volume includes the same information as that for the original volume. If a duplicate volume exists, the display may include the total tape padding blocks that were duplicated for the last duplication, duplication start and end times, total duplication time, and duplicate expiration date.

Note: Duplicate volumes that were created before this release of the EDM software may not display all of this information.

Duplicate Command Options

Table 16-7 lists the **ebreport duplicate** command options and related information.

Table 16-7

ebreport duplicate Command Information

Option	Argument Definition	Description
-active		Lists only the volumes that were last allocated to each active rotation. Also lists the date and time the trail was last written.
-offsite		Lists only the volumes that are marked for offsite storage.
-since <i>date</i>	<i>date</i> is the date for which you wish to view duplication history.	Displays only those volumes for which duplications were attempted on or after the specified date.
-template <i>template name</i>	<i>templatename</i> is the backup template for which you want to display duplication history.	Displays only the volumes that the given template uses. If this option is not supplied or if the given template is "*", all templates appear in the report.
trail <i>trailname</i>	<i>trailname</i> is the backup trail for which you want to view duplication history.	Lists only the volumes that the given trail uses within a trailset. If this option is not supplied or if the given trail is "*", all trails appear in the report.
-trailset <i>trailset name</i>	<i>trailset name</i> is the trailset for which you want to display duplication history.	Displays only the volumes that the given trailset uses. If you use -template , you can specify only "primary" or "alternate;" otherwise, you can specify any valid trailset name. If this option is not supplied or if the given trailset is "*", all trailsets appear in the report.

Sample Backup Duplicate Report

Figure 16-3 shows a sample report using the **ebreport duplicate** command:

edm# **ebreport duplicate**

Figure 16-3

EDM Backup Duplicate Report

edm# **ebreport duplicate**

Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary Trailset

09/15/1998 09:46:51 Rotation ID:A1D7F9BD.71812B77.00000200.F206F11B, 104
backups

Media duplication used on 1 copy

Duplication State: Done, Old, Mode: New

*Orig Vol: A1D7F9BD71812B77 (BDE098) Seq. #: 000025 in TLU: at_dlt_3264_0,
media: DLT tape

Dup Vol: 40D81EE7477F8BDA (BDE146) Seq. #: 000017 in TLU: at_dlt_3264_0, media:
DLT tape

Total Blocks: 349028 Start Time: 09/22/1998 10:54:26 End Time: 09/22/1998
13:06:21

Duration: 001 Hrs. 31 Min., Duplicate Expiration Date: 12/17/1998

09/30/1998 12:57:57 Rotation ID:65D8498A.FD4DC69B.00000200.540819F4, 2 backups

Media duplication used on 1 copy

Backup History Reports

The **ebreport history** command displays reports about the backups that an EDM Backup server and its clients perform. The top line of each report lists the name of the EDM Backup server and the report creation date. Under this line, EDM Backup lists each backup template (schedule) that it backed up, and for each backup template it lists the work items that the template backed up.

For each work item, the report lists the history of backups, one line per backup, with the most recent backups first. The line for each backup includes the time that the backup occurred, the backup level, the backup ID, backup status, the number of files or directories backed up, the backup expiration date, and backup recovery status. (If a backup cannot be recovered, a “NO” appears in the Rcvr field of the history report, which implies that catalog processing needs to be done for that backup.)

If you run the command without any options, the history report can be large. You can use several options to restrict the scope of the report. Use the command options singly or in conjunction with one another to select a restricted set of backups on which to report. The next sections describe the options to the **ebreport history** command.

History Command Options

Table 16-8 lists the **ebreport history** command options and describes the information that is available through the command.

Table 16-8

ebreport history Command Information

Option	Argument Definition	Description
-client <i>clientname</i>	<i>clientname</i> is the client whose backup history you want to display.	Displays all of a client's work items that are backed up to the EDM Backup server.
-workitem <i>workitemname</i> or -item <i>workitemname</i>	<i>workitemname</i> is the client's work item for which you want to display backup history.	Displays a single work item's most recent backup history.
-template <i>templatename</i>	<i>templatename</i> is the backup template for which you want to display backup history.	Displays all of the client work items that were backed up by the backup template (schedule).
-since <i>date [time]</i> -until <i>date</i>	<i>date</i> is the date in the format <i>mm/dd/yy</i> . <i>time</i> (optional) is the time in the format [<i>hh:mm:ss</i>].	Limits the backup history display to a range of dates. Use -since to show the backups that occurred on or after a particular date or use -until to display the backup history that occurred on or up to a particular date.
-trailset <i>primary alternate</i>		Displays the work items that were backed up by a primary or alternate set of media (trailset).
-level <i>levelnumber</i>	<i>levelnumber</i> is a level from 0 to 9, or a range of levels (for example, 0-8), for which you want to display backup history.	Displays all backups of the specified levels unless you use the other options to limit the report coverage.
-recent		Displays all of the work items that were backed up by EDM Backup, from the second most recent level 0 backup to the present. This report lists the standard ebreport history information.

Table 16-8 **ebreport history Command Information (Continued)**

Option	Argument Definition	Description
-volumes or -media		Displays the media volume names that are required to restore these backups.
-recover_size		Displays the amount of disk space that is required to restore each listed backup. The size is listed in KB, MB, GB, or TB as appropriate.
-seconds		Displays seconds in time reported.
-ebimport		Displays only backups that require ebimport(1m) before backup can be restored.
-expire_times		Displays all expire times (catalog, saveset, media), not just the one closest to expiration.
-completeness		Displays backup completeness mode for each listed backup saveset.
-dir		Displays the EBFS directory ID for each listed backup saveset.
-a or -baseline		Includes a baseline backup report after the history report.
-help		Displays command line options for ebreport history .
-nopartials		Skips backups that failed or are in progress.
-size		Displays the amount of data that was actually backed up.
<p>Note: The backup size that this report provides may be inconsistent with the recovery summary size. The algorithms that are used to calculate recovery size and backup size are different.</p>		

Sample Backup History Report

Figure 16-4 shows a sample report that is generated by:

```
# ebreport history -recent
```

for an EDM called fig that backs up two templates (schedules): Generic and Server. The Server Template uses both a Primary media set (trailset) and an Alternate.

Figure 16-4

EDM Backup History Report

```
EDM Backup History Report for server edm at Sept 30 14:19:52 1998
Report options: -recent
**** Work Items for Template Generic, Primary Trailset ****
**Item "vigo:/work" for client "vigo"
Time           Lvl ID           Status    Entries Expires Rcvr
10/19/98 16:12  0  72768A4B.32F65536 complete      210/20/98
10/13/98 16:04  0  72768A4B.32F65406 complete      210/14/98

:**** Work Items for Template Server, Primary Trailset ****
**Item "fig:/" for client "fig"
Time           Lvl ID           Status    Entries Expires Rcvr
10/14/98 16:12  9  72768A4B.3329BF58 unsorted     405310/15/98 NO
10/ 6/98 10:40  0  72768A4B.331EE55C complete     525010/ 7/98
10/12/98 23:36  0  72768A4B.33029A9D complete     500710/17/98
.
.
**** Work Items for Template Server, Alternate Trailset ****

**Item "fig:/" for client "fig"
Time           Lvl ID           Status    Entries Expires Rcvr
10/19/98 0:39  9  72768A4B.330A9265 complete 5045  10/23/98
10/13/98 16:35  0  72768A4B.33038955 complete 5006  10/17/98
.
.
10/30/98 20:00  0  72768A4B.32C9B7ED complete 432710/30/98
```

All backups with complete and delta listings are available for restoring and appear in the EDM Restore window.

Backup Disaster Reports

At the completion of every LOCAL_DATABASE backup (the backup of the EDM Backup database), the script `/usr/epoch/EB/config/local_db_cleanup` automatically generates a minimal disaster report. By default, this report is e-mailed to all EDM Backup administrators, appended to `/usr/epoch/EB/config/disaster-report.log`, and printed to the default system printer.

The minimal disaster report provides the essential information you need to perform a disaster recovery on the server. It is a subset of the full disaster report which is generated by the command **ebreport disaster**. You should run a full disaster report once every backup rotation and whenever significant system changes are made.

Refer to Chapter 19 “Being Prepared for a System Disaster” for instructions on preparing for a disaster. See Chapter 20 “Recovering a Server from a Disk Failure” and Chapter 21 “Recovering a UNIX Client from Disk Failure” for instructions on recovering a server and a client.

Figure 16-5 shows selections from the sections of the EDM Backup FULL disaster report. It contains the following sections:

- Local database volumes report
- Media report
- Backup history report
- Baseline backup history report (for HSM systems)
- Backup coverage report
- Backup installation report
- Backup configuration files
- List of installed clients
- Library Manager configuration data
- Filesystem table (`/etc/vfstab`)
- Locally mounted disks
- HSM local configuration
- The root crontab file

Figure 16-5**EDM Backup FULL Disaster Report**

EDM Backup FULL Disaster Report for server "bilbo" on Sept 8 11:16:16 1998

LOCAL_DATABASE Backup Volumes Report

**Local
Database
Volumes
Report**

The following volumes contain the most recent LOCAL_DATABASE backup which will be required in the event of a Disaster Recovery:

Saveset ID 7271F980.2FAD095E for LOCAL_DATABASE backup on 9/7/98 13:54,

bilbo_pri_dlt #0012 (50BE6FE05346D06C) - currently in library unit "at_dlt_3264_0", slot #7

bilbo_pri_DLT #0013 (EBDD020907E7982C) [duplicate] - currently in library unit "at_dlt_3264_0", slot #3

This LOCAL_DATABASE backup will require 40.9 MB of disk space to be recovered.

EDM Backup Media Report for server bilbo on Sept 8 11:16:17 1998

Report options: none

Summary of all media, listed by media rotation groups

**Media
Report**

Rotations for Template "bilbo", Trail "bilbo_pri_dlt", Primary Trailset

08/24/98 14:06 Rotation ID:E1BE6F9E.22C1253E.00000200.A804A0B9, 88 backups, Media duplication not used

*Vol ID: 50BE6FE05346D06C, media: DLT tape, number 0012

Vol ID: E1BE6F9E22C1253E, media: DLT tape, number 0011

Rotations for Template "argon", Trail "argon_alt_dlt", Alternate Trailset

08/22/98 20:03 Rotation ID:34BE6662.C1CEE018.00000200.48080A35, 1 backup, Media duplication not used

*Vol ID: 34BE6662C1CEE018, media: DLT tape, number 0009

Vol ID: E1BE6F9E22C1253E, media: DLT tape, number 0011

Rotations for Template "argon", Trail "argon_alt_dlt", Alternate Trailset

08/22/98 20:03 Rotation ID:34BE6662.C1CEE018.00000200.48080A35, 1 backup, Media duplication not used

*Vol ID: 34BE6662C1CEE018, media: DLT tape, number 0009

.
.
.

EDM Backup History Report for server bilbo at March 8 11:16:21 1998
Report options: -recent -completeness -dir

History Report

**** Work Items for Template argon, Primary Trailset ****

**Item "argon" for client "cheetah"

Time	Lvl	ID	Status	Entries	Expires	serverdb	Completeness	Directory ID
9/ 7/98 17:58	0	7271F980.2FAD0B70	complete	33075	9/12/98	normal	files	
BDBE8F48.D50C5B38.0023CD00.F40D4A0A								
9/ 6/98 17:58	9	7271F980.2FABB9E6	delta	4	9/11/98	normal	files	
BDBE8F48.D50C5B38.0022D400.610AF0E0								
9/ 5/98 17:58	9	7271F980.2FAA685E	delta	24	9/10/98	normal	files	
BDBE8F48.D50C5B38.0021C000.B20A1983								
9/ 4/98 17:58	9	7271F980.2FA916DA	delta	1006	9/9/98	normal	files	
BDBE8F48.D50C5B38.0020B300.FF0C66BE								
9/ 3/98 17:58	8	7271F980.2FA7C582	delta	24	9/8/98	normal	files	
BDBE8F48.D50C5B38.001F4E00.FC0D0D5F								

**** Work Items for Template argon, Alternate Trailset ****

**Item "argon" for client "cheetah"

Time	Lvl	ID	Status	Entries	Expires	serverdb	Completeness	Directory ID
9/22/98 20:02	0	7271F980.2F999998	complete	32434	9/24/98	normal	files	
34BE6662.C1CEE018.00000500.4B0EB453								
.								
.								
.								

EDM Backup Baseline Backup History Report for server fig at Sept 27 14:43:49 1998
Report options: none

**** Baseline Backups for Template Server, Primary Trailset ****

**Item "fig:/catalogs"

Time	Lvl	ID	Status
9/ 5/98 10:23	B1	72768A4B.32F8A612	partial

Baseline Backup History Report (for HSM systems)

**Item "fig:/client_data1"

Time	Lvl	ID	Status
9/ 5/98 17:41	B1	72768A4B.32F90C9A	no cat
9/ 5/98 10:23	B1	72768A4B.32F8A611	no cat
9/ 4/98 19:56	B1	72768A4B.32F7DADA	partial

EDM Backup Coverage Report for server adam at Sept 8 11:16:23 1998

Coverage Report

Report options: none

Filesystems currently backed up:CurrentMaxCurrentMax

```
-----
adam:/ 4381/41536 files,50.3 MB/ 74.9 MB adam:/data1 27/ 63152 files,3.4 MB/ 250.0
MB adam:/data2 21/ 63152 files, 3.3 MB/250.0 MB
hamster:/ 17854/215040 files,600.1 MB/ 778.2 MB negril:/ 3685/ 98176 files, 64.1
MB/ 187.9 MB negril:/data 5/ 191872 files, 1.3 MB/ 750.8 MB
..
Total: 17 filesystems backed up 62480/ 2202176 files, 1.4 GB/ 6.9 GB
```

EDM Backup Installation Report for server adam at Sept 27 14:23:26 1998

Report options: -all

.
.
.

Installation Report

EDM Backup currently running load 6.0.0.0

/usr/epoch IS A SYMLINK to /ep_usr/epoch

/usr/epoch/EB is a real directory under /ep_usr/epoch

/usr/epoch/GENDIR IS A SYMLINK to /home/epoch

/usr/epoch/EB/adam is a real directory under /usr/epoch/EB

/usr/epoch/EB/bin is a real directory under /usr/epoch/EB

/usr/epoch/EB/catalogs IS A SYMLINK to /home/epoch/EB/catalogs

.
.

The local client is of the type: sun_sun4_v55_srv

The client backup username is: ebadmin

The user ID for ebadmin is: 24375

The group ID for ebadmin is: 25

The home directory for ebadmin is: /usr/epoch/EB

Client negril is of type sun_sun4_v55_emc (5.0.0.0) installed 1998 Sept 13 16:06:21

Client adam is of type sun_sun4_v55_srv (5.0.0.0) installed 1998 Sept 27 13:24:46

Client hamster is of type hp_700_v9 (5.0.0.0) installed 1998 Sept 27 13:55:29

End of EDM Backup Installation Report for server adam at Sept 27 14:23:26 1997

Displaying current EDM Backup configuration...

Sept 8 11:16 1998 /tmp/eb.cfg Page 1

ebserver: "bilbo"

{

client backup username: "ebadmin";

backup administrator usernames:

"root",

"gil";

authorized backup list:

"argon",

"bilbo",

.

.

.

startup parameters:

{

perform initial full backup as soon as possible;

} /* end startup parameters */

} /* end server block */.

.

.

.

#Clients Installed File

#

#Format: I,len,host,timestamp,len,method,invocations,platform,RMS,vlength,version,;

#

I, 7,wildcat,837787783, 7,edmlink,0,99,0, 7,5.0.0.0,;

I, 4,fish,842992815, 7,netware,10,113,0, 7,5.0.0.0,;

I, 6,berlin,850748622, 7,netware,10,53,0, 7,5.0.0.0,;

I, 7,warthog,850776740, 7,netware,0,113,0, 7,5.0.0.0,;

I, 4,zero,850777296, 7,netware,0,114,0, 7,5.0.0.0,;

I, 7,hamster,851018343, 3,rsh,0,75,0, 7,5.0.0.0,;

I, 6,jumper,856192150, 7,edmlink,0,107,0, 7,5.0.0.0,;

I, 4,vigo,856385813, 7,edmlink,0,109,1, 7,5.0.0.0,;

I, 8,chipmunk,857155162, 3,rsh,1,97,0, 7,6.0.0.0,;

I, 6,negril,858032953, 7,edmlink,0,109,1, 7,6.0.0.0,;

I, 12,indianapolis,858725141, 7,netware,1,53,0, 7,6.0.0.0,;

I, 3,fig,859312730, 6,direct,0,108,1, 7,6.0.0.0,;

I, 6,bolton,859571785, 7,edmlink,0,93,0, 7,6.0.0.0,;

I, 7,pilgrim,859821588, 7,edmlink,0,99,0, 7,6.0.0.0,;

**Backup
Configuration
Files**

**Installed
Clients**

Displaying library manager configuration (used with lmconfig...)

Library Manager Configuration

* Lu_name	Name	ID	Status
L offline_0	-	-	synced
L offsite_0	-	-	synced
L at_dlt_3264_0	-	(0,1,1,0)	synced
D at_dlt_3264_0	drive_0	(0,1,5,0)	enabled
D at_dlt_3264_0	drive_1	(0,1,4,0)	enabled
D at_dlt_3264_0	drive_2	(0,1,3,0)	enabled
L hp_mf_c17xx_0	-	(0,2,6,0)	synced
D hp_mf_c17xx_0	drive_0	(0,2,5,0)	enabled
D hp_mf_c17xx_0	drive_1	(0,2,4,0)	enabled
D hp_mf_c17xx_0	drive_2	(0,2,2,0)	enabled
D hp_mf_c17xx_0	drive_3	(0,2,1,0)	enabled

Filesystem Table /etc/vfstab

Sept 21 10:51 1998 /etc/vfstab Page 1

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
#						
#/dev/dsk/cld0s2	/dev/rdisk/cld0s2	/usr	ufs	1	yes	-
fd	- /dev/fd	fd	-	no	-	
/proc	- /proc	proc	-	no	-	
/dev/dsk/c0t3d0s1	- -	swap	-	no	-	
/dev/dsk/c0t3d0s0	/dev/rdisk/c0t3d0s0	/	ufs	1	no	-
/dev/dsk/c0t3d0s6	/dev/rdisk/c0t3d0s6	/usr	ufs	2	no	-
/dev/dsk/c0t2d0s3	/dev/rdisk/c0t2d0s3	/ep_usr	ufs	3	yes	-
/dev/dsk/c0t3d0s5	/dev/rdisk/c0t3d0s5	/home	vxfs	4	yes	-
/dev/dsk/c0t2d0s0	/dev/rdisk/c0t2d0s0	/data1	vxfs	5	yes	-
/dev/dsk/c0t2d0s1	/dev/rdisk/c0t2d0s1	/data2	vxfs	6	yes	-
/dev/dsk/c0t3d0s7	/dev/rdisk/c0t3d0s7	/home1	ufs	7	yes	-
swap	- /tmp	tmpfs	-	yes	-	

Displaying locally mounted disks...

/	((dev/dsk/c0t3d0s0)):	8192 block size 1024 frag size	
153534 total blocks	50292 free blocks	34952 available	41536 total files
37126 free files	8388632 filesys id		
ufs fstype	0x00000004 flag	255 filename length	Locally Mounted Disks
/usr	((dev/dsk/c0t3d0s6)):	8192 block size 1024 frag size	
769694 total blocks	432834 free blocks	355874 available	192576 total files
177771 free files	8388638 filesys id		
ufs fstype	0x00000004 flag	255 filename length	
.			
.			
/home	((dev/dsk/c0t3d0s5)):	8192 block size 1024 frag size	
495936 total blocks	176534 free blocks	150656 available	23840 total files
21507 free files	8388637 filesys id		
vxfs fstype	0x00000004 flag	255 filename length	

EpochMigration Local Configuration

EpochMigration System Configuration:

Enable_stage_out	Max_trails	Enable_self_describing
Y	1	N

HSM Local Configuration
(HSM is optional with EDM)

Staging trail "PubsTrail_1"

Stage outs enabled: Y Media: EO Unrestricted
Self-Describing enabled: N

Enable	HWM	LWM	PSWM	Delay	Mntpoint
Y	95	88	80	0	defaults for PubsTrail_1
Y	95	88	80	0	/home

.

EpochMigration Current Migration Volumes

Current primary staging volumes are:

Staging trail "PubsTrail_1"
Sequence: 14 Mside: 1 Valid: 11CC39F59912E6A9 Nblocksavail: 314529

Staging trail "PubsTrail_2"
Sequence: 13 Mside: 1 Valid: 89CC2F69030965E2 Nblocksavail: 314529


```

Displaying root crontab...
#ident"@(#)root1.1193/04/08 SMI"/* SVr4.0 1.1.3.1*/
#
# The root crontab should be used to perform accounting data collection.
#
0 2 * * 0,4 /etc/cron.d/logchecker
5 4 * * 6 /usr/lib/newsyslog
15 3 * * * /usr/lib/fs/nfs/nfsfind
# Invoke EDM Backup backup program #EPCebs
45 13 * * * /usr/epoch/EB/bin/ebbackup bilbo >/dev/null 2>&1 #EPCebs
# Invoke EDM Backup backup program #EPCebs
00 14 * * * /usr/epoch/EB/bin/ebbackup argon >/dev/null 2>&1 #EPCebs
# Invoke EDM Backup backup program #EPCebs
15 14 * * * /usr/epoch/EB/bin/ebbackup lucifer >/dev/null 2>&1 #EPCebs
# Invoke EDM Backup catalog expiration program #EPCebs
30 00 * * * /usr/epoch/EB/bin/ebexpire -expire -purge >/dev/null 2>&1 #EPCebs
# Invoke EDM Backup catalog cleanup program #EPCebs
00 1 * * * /usr/epoch/EB/bin/ebcatclean -fix_saveset >/dev/null 2>&1 #EPCebs
# Invoke EDM Backup LOCAL_DATABASE validity check program #EPCebs
00 3 * * * /usr/epoch/EB/config/local_db_warning >/dev/null 2>&1 #EPCebs
#40 * * * * /usr/epoch/lib/epnewlog 500000 > /dev/null 2>&1#EPCgl
#00 23 * * 6 /usr/epoch/lib/epnewlog > /dev/null 2>&1#EPCgl
#00 07 * * * /usr/epoch/lib/eptrunclog root > /dev/null 2>&1#EPCgl
#30 08 * * * /usr/epoch/lib/epcleanup > /dev/null 2>&1#EPCgl
40 * * * * /usr/epoch/lib/epnewlog 500000 > /dev/null 2>&1#EPCgl
00 23 * * 6 /usr/epoch/lib/epnewlog > /dev/null 2>&1#EPCgl
00 07 * * * /usr/epoch/lib/eptrunclog root > /dev/null 2>&1#EPCgl
30 08 * * * /usr/epoch/lib/epcleanup > /dev/null 2>&1#EPCgl
#50 8 * * * /usr/epoch/lib/ebfs/ebfs_cleanup > /dev/null 2>&1#EPCebfs
50 8 * * * /usr/epoch/lib/ebfs/ebfs_cleanup > /dev/null 2>&1#EPCebfs
.
.
.
End of EDM Backup FULL Disaster Report for server "bilbo" on Sept 8 11:16:16 1998

```

End of EDM Backup Disaster Report

Backup Baseline Reports

In HSM systems, **ebreport baseline** generates the baseline report which presents a summary of backup baseline activity as reflected in the saveset database. A status line is printed for every non-expired baseline backup of every work item selected by the arguments.

Table 16-9

ebreport baseline Command Information

Option	Argument Definition	Description
-client <i>clientname</i>	<i>clientname</i> is the client whose baseline history you want to display.	Displays only backups that were created for the named client.
-item <i>workitemname</i>	<i>workitemname</i> is the client's work item for which you want to display baseline history.	Displays only backups that were created for the named work item. If the name "*" is used, all work items are selected. Note that "*" must be quoted on the command line.
-template <i>templatename</i>	<i>templatename</i> is the backup template for which you want to see a baseline summary.	Displays only backups that were run from the named template (schedule). If the name "*" is used, all work items are selected. Note that "*" must be quoted on the command line.
-completeness		Also displays the backup completeness mode for each reported work item backup.
-recent		Lists only baseline backups since the second most recent level 0 backup for each work item.
-since <i>date</i> [<i>time</i>]	<i>date</i> is the date in the format <i>mm/dd/yy</i> . <i>time</i> (optional) is the time in the format [<i>hh:mm:ss</i>].	Limits the backup display to a range of dates. Use -since to show the backups that occurred on or after a particular date or use -until to display the backup that occurred on or up to a particular date.
-until <i>date</i>		

Table 16-9 **ebreport baseline Command Information (Continued)**

Option	Argument Definition	Description
-trailset primary alternate		Displays the work items that were backed up by a primary or alternate trailset.
-level [B1 B2]		Displays only savesets for backups of the given level. You can enter up to two -level options in a single invocation, each occurrence adding another level to the selection set.

Figure 16-6 shows a sample baseline report generated by **ebreport baseline**.

Figure 16-6

EDM Backup Baseline Report

```

EDM Backup Baseline History Report for server tesla at
Sept 17 09:37:43 1998
Report options: none

Template Name
*** Baseline Backups for Template bline_bt_1, Primary Trailset ***

**Item "bline_wi_1"
Time           Lvl ID           Status
9/16/98 18:00 B1  55412298.2FB92071 no cat

*** Baseline Backups for Template default, Primary Trailset ***

Work Item Name
**Item "tesla:/data5"
Time           Lvl ID           Status
Baseline Level 9/16/98 18:00           B1  55412298.2FB9206C no cat
                9/15/98 18:00           B1  55412298.2FB7CEFA no cat
                9/14/98 18:00           B1  55412298.2FB67D77 no cat
                9/13/98 18:00           B1  55412298.2FB52BF5 no cat
Saveset ID     9/12/98 18:00           B1  55412298.2FB3DA6C no cat

Backup status

```

Backup Completion Reports

EDM Backup prepares backup completion reports and can send them to specified individual(s) via a shell script. (For setup details see "Backup Completion Script" on page B-79.) Figure 16-7 shows a sample backup completion report.

Figure 16-7

EDM Backup Completion Report

```

Date, Time and 9/05/98 18:31:21 [ 2295:/usr/epoch/EB/bin/ebbackup]
Process ID #, Summary report for processing template "mwf"
Template

Date, Time and 9/05/98 18:31:21 [ 2295:/usr/epoch/EB/bin/ebbackup]
process ID #, processing of work item "cad1-all" via template "default" Level 0
Work Item, SUCCEEDED
Template, trailset was "cdr", trail was "cdr tape", 59 files backed up in
Trailset, Trail, 886KB
Number of Files 9/05/98 18:31:21 [ 2295:/usr/epoch/EB/bin/ebbackup]
Backed up and processing of work item "cad2-all" via template "default"
Number of kb SUCCEEDED
Used trailset was "cdr", trail was "cdr tape", 312 files backed up
in 55809KB
9/05/98 18:31:21 [ 2295:/usr/epoch/EB/bin/ebbackup]
processing of work item "cad3-all" via template "default"
SUCCEEDED
trailset was "cdr", trail was "cdr tape", 10257 files backed up
in 135514KB
9/05/98 18:31:21 [ 2295:/usr/epoch/EB/bin/ebbackup]
processing of work item "cad4-all" via template "default"
SUCCEEDED
trailset was "cdr", trail was "cdr tape", 25306 files backed up
in 203749KB

```

The server **eb_server_config** installation procedure creates the **mailok** script to which it passes the backup completion information. The script mails a completion statement to individuals responsible for backup operations, and/or writes them to a log.

Because the **ebbackup** program mails these reports immediately after a backup, you can read them as soon as the backup completes.

Backup Failure Reports

Whenever EDM Backup encounters an error that prevents backup completion (for example, a client system has crashed), it generates a backup failure report. (For setup details see “Backup Failure Script” on page B-80.)

The EDM Backup program can send backup failure reports to specified individuals via a shell script. Because EDM Backup mails these reports whenever a failure occurs, you are notified of a failure as soon as it happens. On the other hand, if you don't receive one of these reports, you can assume your backups are successful. When you receive a backup failure report, you should fix the problem with the client system. However, EDM Backup continues to back up all other clients in the backup template (schedule), skipping those that had a problem.

Figure 16-8 shows a sample backup failure report.

Figure 16-8

EDM Backup Failure Report

Date, Time, Process ID #, Error Number, Work Item Name, and Reason for Failure	9/06/98 06:22:21 [3423:ebbackupd errno=35{ Operation would block} , ec=0x19] Workitem "doc1-all" backup TIMED-OUT
---	---

The server **eb_server_config** installation procedure creates the **mailerr** script to which it passes the backup failure information. The script mails a failure statement to individuals who are responsible for backup operations, and/or writes them to a log.

Backup Coverage Reports

The **ebreport coverage** command makes it easy for you to determine if new filesystems were added to client systems and if they are getting backed up. When the report lists a filesystem that EDM Backup is not currently backing up, and the filesystem is one that you want to backup, you'll need to edit the client's work item statement to add the filesystem to the list of backup files.

Note: **ebreport coverage** reports on Unix and Windows NT filesystems only (no NetWare filesystems).

You can use the **ebreport coverage** command to display backed up and non-backed up filesystems on EDM Backup clients. The **ebreport coverage** command displays the backup status of all filesystems or you can use the options to display the following information.

Table 16-10

ebreport coverage Command Information

Command	Argument Definition	Description
-client <i>clientname</i>	<i>clientname</i> is the client whose backup history you want to display.	Displays a single client's backed up and non-backed up filesystems.
-completeness		Shows what kind of data is being backed up (for resident files only).
templatename	<i>templatename</i> is the backup template (schedule) for which you want to display backup history.	Displays a backup template's non-backed up filesystems; with the <i>templatename</i> option displays the backup status of the specified template(s') filesystems.
-installed		Shows installed EDM Backup clients – displays all backed up and non-backed up filesystems in installed client list.

Figure 16-9 shows a report generated by **ebreport coverage** for three clients (adam, hamster, and negril), and identifies the fields in the report.

Figure 16-9

EDM Backup Coverage Report

EDM Backup Coverage Report for server adam at Sept 8 11:16:23 1998

Report options: none

Filesystems currently backed up:	Current	Max	Current	Max
adam:/	4381/	41536 files, 50.3 MB/	74.9 MB	
adam:/data1	27/	63152 files, 3.4 MB/	250.0 MB	
adam:/data2	21/	63152 files, 3.3 MB/	250.0 MB	
hamster:/	17854/	215040 files, 600.1 MB/	778.2 MB	
negril:/	3685/	98176 files, 64.1 MB/	187.9 MB	
negril:/data	5/	191872 files, 1.3 MB/	750.8 MB	
negril:/data1	4/	191872 files, 1.3 MB/	750.8 MB	
.				
.				
.				
Total: 17 filesystems backed up	62480/	2202176 files, 1.4 GB/	6.9 GB	

Volume Reports

The **dbreport *reportname*** command generates reports from the system administration database.

Note: Ordinarily, only privileged users (those running as root) can run **dbreport**.

Some of these reports are listed in Table 16-11, to see a full list of reports see the **dbreport** man page.

Table 16-11**dbreport Command Information****Report Name Description**

volume	<p>Generates a report of all the volumes known to the system.</p> <p>The fields of the report describe the type of media, the name of the application that currently owns the volume, the name assigned to the volume, the sequence number of the volume, the side of the volume, and the barcode of the volume, if any.</p>
available	Generates a report of just those volumes that are currently available for allocation.
appl_usage	<p>Generates a report of application volume usage statistics.</p> <p>The fields of the report are the application name, the volume name, the media type, the sequence number, the side, barcode, the number of blocks available, used, and stale in 1 KB units, the percentage of the volume which is stale data, and the number of files used and stale on the volume.</p>
online	<p>Generates a report of all media in system library units.</p> <p>The report is sorted by application, media type and application-dependent name.</p>
offline	<p>Generates a report of all media not in any system library unit.</p> <p>The report is sorted by application, media type and application-dependent name.</p>
offsite	<p>Generates a report of all media which has been moved to offsite storage.</p> <p>The volumes in this category can be re-introduced to the system by being injected into a library unit.</p> <p>The report is sorted by application, media type and application-dependent name.</p>

Table 16-11

dbreport Command Information (Continued)

Report Name	Description
hsm	<p>Generates a report of staging volume usage statistics.</p> <p>The fields of the report are the volume name, sequence number, side, barcode, the number of blocks used and stale in 1KB units, the percentage of the volume which is stale data, and the number of files used and stale on the volume.</p>
baseline	<p>Generates a report of baseline volume usage statistics.</p> <p>The fields of the report are the volume name, sequence number, side, barcode, the number of blocks used and stale in 1KB units, the percentage of the volume which is stale data, and the number of files used and stale on the volume.</p>
compaction	<p>Generates a report you can use to estimate the staleness of volumes in order to determine which HSM volumes to compact with the emcompact utility.</p>

Log Files

You can access backup log files directly and monitor them or review them for troubleshooting. For example, use **tail -f** to monitor progress during processing and use **vi** or other editor to review logs at a later time.

When filled, the oldest ten percent of these files is deleted on an ongoing basis.

Server Log Files

EDM Backup automatically creates log files in the directory `/usr/epoch/EB/log` on the EDM Backup server.

- The `backups.log` file contains information about backup operations. EDM Backup adds information to this file each time it backs up a template's work items. Selected notifications that appear in this log file also appear in other backup reports. It accumulates detailed shutdown and startup information each time a database work item is backed up.

Every two minutes **ebbackup** reports the average rate in KB/s for ALL work items being backed up. This rate is affected by process timing, data buffering, and overhead in **ebbackup**. If you want to see the rate for a specific tape drive, see the EDM Library Unit Manager window which reports on an active drive every thirty seconds.

- The `recoveries.log` file contains file restore startup and completion notifications. Use these files for comprehensive information about backup and restore activities on your EDM server.
- The `ebcat.log` contains startup and shutdown times for catalog processing and output from **ebexpire** and **ebimport**.
- The `template_name.log` records backup information for a single template.

Whenever you want to see the backup history of a single backup template (schedule), use the `template_name.log` file. The information in this log file varies depending on the *logging level* you specified in the configuration database (see “Server Log File” on page B-78). Thus, you can use the file to view a history of backup-related events for a single template.

The default log file is located in
`/usr/epoch/EB/log/default_template.log`.

The default logging level, *stats*, reports when each client backup or restore begins, and includes periodic progress indications.

There are five logging levels. Use the *debug* and *per file* levels to diagnose problems only when instructed by customer service.

Local Client Log Files

EDM Backup automatically creates two log files on the EDM Backup client: the `backups.log` and `recoveries.log` in the directory `/usr/epoch/EB/CLIENT_HOME/client`. The `backups.log` file contains an audit trail of backup-related activities listed in chronological order. The `recoveries.log` file contains an audit trail of restore-related activities listed in chronological order.

- `backups.log` accumulates detailed scanning information each time a local work item is backed up.
- `recoveries.log` records general start and end notifications for restore processing each time a local work item is restored.

Remote Client Log Files

On the remote clients, `backups.log` and `recoveries.log` files reside in the directory `/usr/epoch/EB/CLIENT_HOME/clientname`. They record network backups and restore operations.

Other Logs

Other logs record volume management and other system activity:

- System logs are located in `/var/adm`
- System logs are archived in `/usr/epoch/adm`
- Circular logs are located in `/usr/epoch/adm/circular`

System logging is configurable in `/etc/syslog.conf`.

Part IV

Command Line Interfaces

17 **Configuring Library Managers**

When you change EDM configuration by adding or removing a library unit, you need to reconfigure the software to recognize the change.

This chapter describes the script that you use to install device drivers and configure Library Managers for library units that are connected to the EDM.

The chapter describes the following tasks:

- Using the lmconfig Utility
- Listing Library Managers
- Installing Device Drivers
- Updating Device Drivers
- Removing Device Drivers
- Configuring a Library Manager
- Deconfiguring a Library Manager

Using the Imconfig Utility

The **lmconfig** utility, enables you to:

- list currently configured Library Managers
- install, update, and remove device drivers
- configure and deconfigure Library Managers
- access a help option that briefly describes the main menu entries

(Refer to the lmconfig man page for more information about this utility.)

Note: lmconfig is located in /usr/epoch/bin. Make sure that this pathname is defined in your PATH environment variable.

To start the configuration script, log in as root and enter the following command to display the main menu. In this menu, you select the configuration you want to perform.

```
# lmconfig
```

```
EMC  LIBRARY  MANAGER  CONFIGURATION  TOOL
```

```
Main Menu
```

```
1 LIST           current Library Manager configurations
2 INSTALL        EMC drivers
3 UPDATE         EMC drivers
4 REMOVE         EMC drivers
5 AUTOCONFIG     Automatically configure all library unit
6 DECONFIGURE    a Library Manager
7 HELP
```

```
Choose the configuration operation you want (1,2,3,4,5,6,7,q)
```


Listing Library Managers

When you choose 1 LIST from the main menu, Library Managers that are currently configured in /usr/epoch/etc/lm appear. lmconfig lists the name of the Library Manager and the device's SCSI address for the robot and each drive, as shown in the following example:

lmconfig completed configurations:

```
offline_0
offsite_0
at_452_0
  r0: 0 2 2 0
  d0: 0 2 3 0
  d1: 0 2 4 0
  d2: 0 2 5 0
  ...
```

Library Manager Name

The Library Manager's name identifies the manufacturer, drive type, and model number of the device.

Note: In releases previous to EDM 4.5.0, the Library Manager name includes the drive type (DLT, DTF, HITC, etc.); for example, "at_dlt_452_0."

For example, the Library Manager name, at_452_0, has the following meaning:

Table 17-1

at	Manufacturer of the automated tape library unit: ATL Products.
452	Manufacturer's model number; in this example, the ACL 4/52 automated tape library unit. (4 drives/52 slots)
_0	Indicates the first Library Manager that is configured for this library unit type. The suffix increments for each additional library unit of this type that you configure.

SCSI Address

The SCSI address includes the system board number, SCSI bus, SCSI target ID, and logical unit number (LUN) of the library unit robot and drive(s); for example:

```

                                r0:    0 2 2 0
                                d0:    0 2 3 0
System Board # _____|_|_|
SCSI Bus _____|_|_|
SCSI Target ID _____|_|_|
LUN _____|_|_|

```

In this example, the library unit's robot and drive are on system board 0, SCSI bus 0; the robot's SCSI target ID is 0, and its LUN is 0. The library unit has one internal drive at SCSI target ID 1, LUN 0.

Installing Device Drivers

When you choose 2 INSTALL from the main menu, lmconfig installs the device drivers into the /devices directory. You must install device drivers before you configure a Library Manager. This option requires that at least one library unit be connected to the server and operational.

To install device drivers, use the following procedure.

1. Choose 2 INSTALL from the main menu. A prompt asks you to confirm the installation:

```

Main Menu
  1 LIST          current Library Manager configurations
  2 INSTALL      EMC drivers
  3 UPDATE        EMC drivers
  4 REMOVE        EMC drivers
  5 AUTOCONFIG    Automatically configure all library units
  6 DECONFIGURE   a Library Manager
  7 HELP
Choose the configuration operation you want (1,2,3,4,5,6,7,q)? 2
About to install all EMC drivers.
Do you wish to continue (y,n)? y

```

2. Enter **y** to begin driver installation. (Note that driver names vary by platform.) The script displays several messages that confirm driver installation.

```

Modifying kernel driver.conf files
Modifying /kernel/drv/st.conf

Installing drivers
Installing driver mo
Driver mo installed
Installing driver sjb
Driver sjb installed

```

Note: Ignore messages that indicate failure of mo driver installation.

3. After the drivers are installed, shut down the system to the PROM level by entering the following command:

```
# shutdown -y -i6 -g0
```

This command enables a reconfiguration reboot of the server.

4. After EDM shuts down and reboots, log in as root and restart lmconfig.

The main menu appears, as shown:

```
EMC  LIBRARY  MANAGER  CONFIGURATION  TOOL
```

```
Main Menu
```

```
1 LIST          current Library Manager configurations
2 INSTALL       EMC drivers
3 UPDATE        EMC drivers
4 REMOVE        EMC drivers
5 AUTOCONFIG    Automatically configure all library uni
6 DECONFIGURE   a Library Manager
7 HELP
```

```
Choose the configuration operation you want (1,2,3,4,5,6,7,
```

EDM probes the bus for all attached hardware and assigns device nodes in the filesystem that represent the devices that are found. It also configures the logical namespace in /dev and the physical namespace in /devices.

5. Select 5 AUTOCONFIG to configure Library Managers automatically for the attached library units. Refer to “Configuring a Library Manager” on page 17-8 for this procedure.

Updating Device Drivers

Choose 3 UPDATE from the lmconfig menu to reinstall device drivers. You must update the device drivers after updating the EDM software or updating the module that contains the drivers.

When you update device drives, no Library Manager reconfiguration is required.

To update the device drivers:

1. Choose 3 UPDATE from the main menu.
2. Confirm the update at the prompt.

Removing Device Drivers

Choose 4 REMOVE from the **lmconfig** menu to remove all device drivers from the /devices directory. You must remove device drivers before deinstalling the EDM software.

To remove device drivers, select 4 REMOVE from the main menu. Then confirm the removal at the prompt.

Sample output appears below.

Main Menu

```
1 LIST          current Library Manager configurations
2 INSTALL      EMC drivers
3 UPDATE       EMC drivers
4 REMOVE       EMC drivers
5 AUTOCONFIG   Automatically configure all library units
6 DECONFIGURE  a Library Manager
7 HELP
```

Choose the configuration operation you want
(1,2,3,4,5,6,7,q)? **4**

About to remove all EMC drivers.

Do you wish to continue (y,n)? **y**

Modifying kernel driver.conf files

Modifying /kernel/drv/st.conf

Removing drivers

Removing driver mo

Removing driver sjb

Configuring a Library Manager

Use the AUTOCONFIG option of `lmconfig` to configure a Library Manager automatically for each library unit that is attached to the EDM.

AUTOCONFIG verifies that offline and offsite daemons are configured. Then it searches for all device nodes in the system, acquires all necessary information, and configures a library manager for each library unit. AUTOCONFIG automatically unloads all drives and puts the media into empty slots.

Because the operation is automatic, you do not have to know the system board numbers, SCSI bus numbers, target IDs, and LUN numbers for each device.

The `lmconfig` utility creates a subdirectory in `/usr/epoch/etc/lm`, copies a sample configuration file into the directory and modifies it, creates a link to the executable file, and adds the pathname of the new Library Manager to the Volume Manager's configuration file. (See Appendix C "Volume Management Configuration Files" for more information about how the Volume Manager uses the `vm.cfg` file.)

Note: You can also use enhanced `lmconfig` to run AUTOCONFIG and tell it not to ask any questions. If it detects any drives with media in them, AUTOCONFIG unloads the media automatically without asking you, and configures all unconfigured library units. You run enhanced `lmconfig` by using the command **`lmconfig -A`**.

Preparing for Configuration

Before you configure Library Manager(s), verify that:

- your hardware configuration is valid
- library units are properly cabled, powered up, and online
- device drivers were installed successfully and EDM rebooted
- all library unit drives are operational
- at least one piece of media is loaded in each library unit

Running lmconfig

To start configuration, do the following:

1. Be sure you are logged in as root.
2. Enter the following command to display the lmconfig main menu:

```
# lmconfig
```

3. Choose 5 AUTOCONFIG from the main menu and then confirm autoconfiguration at the prompt.

```
EMC LIBRARY MANAGER CONFIGURATION TOOL
```

```
Main Menu
```

```

1 LIST          current Library Manager configurations
2 INSTALL      EMC drivers
3 UPDATE       EMC drivers
4 REMOVE       EMC drivers
5 AUTOCONFIG   Automatically configure all library uni
6 DECONFIGURE  a Library Manager
7 HELP
```

```
Choose the configuration operation you want
(1,2,3,4,5,6,7,q)? 5
```

Make sure the following are all true before continuing:

1. The library units were set up, cabled correctly, powered on and online.
2. Drivers have been successfully installed and the system rebooted.
3. All library unit drives are operational.
4. At least one piece of media is in each library unit.
5. The BCS Calypso unit that has the library unit to be configured must not be running any backups or have any opened streams to the drives.

```
Would you like to continue with autoconfiguration (y,n)? y
```

If All Library Units Are Configured

AUTOCONFIG looks for unconfigured devices, active library units, and loaded drives. If all library units are already configured, the following appears:

```
=====
Starting Autoconfig v3.0
=====

Determining unconfigured devices:    100%
Searching for active library units:  100%
_autoconfig failed: *** Error: No non-configured library units found

autoconfig: No configuration done

lmconfig warning: AUTOCONFIG failed
```

Nothing was done because all available library units are already configured.

If Media Is Found In Any Drive

If AUTOCONFIG detects media in a drive it automatically unloads the drive and places the media into an empty slot. (Sample output follows.)

Select **y** (yes) to continue with configuration.

Note: If a mechanical problem does not allow the media to be moved, AUTOCONFIG fails. If you cannot fix the problem, shut off the problem library unit and then run AUTOCONFIG again to configure the remaining library units.


```

=====
Starting Autoconfig v3.0
=====

Determining unconfigured devices:      100%
Searching for active library units:    100%
Checking for loaded drives:            100%

*****
* Found some drives loaded with media.
* Unconfigured drives must not have media in them.
*
* WARNING: The unload program will unload all
*           unconfigured drives attached to this server.
*****

Would you like to unload the drive(s) (y,n)? y

Searching for loaded drives in library unit:
Vendor[ HP] Product[ C1710T] :: Board Bus Target LUN [ 0 4 0 0]

Found 1 drives loaded with media
Unloading.....

Moving media from drive 0 to slot 6

```

The Configuration Process

AUTOCONFIG displays a list of available library units that are not yet configured. At the prompt, enter "a" for all, or a comma-separated list to select some but not all of the listed library units.

AUTOCONFIG now configures all or selected library units.

Please choose which library units to configure :

1.Vendor[HP] Product[C1710T] :: Board Bus Target LUN [0 4 0 0]

Please enter a comma separated list of the library units you would like to configure, or 'a' for all : **a**

::

Getting Info on Library Unit #0 : HP : C1710T : 6.10

The robot on library unit 1 is located at:

=== BOARD:0 BUS:4 TARGET:0 LUN:0 ===

Number of drives = 2

Number of slots = 32

Number of inlets = 1

Media found in slot number 0

Library unit supports drive to drive moves

Using compatible configuration files: hp_c17xx.attr / hp_c17xx.

= = = = =

Loading Drive 0. Please Wait.....

Waiting for drive to be ready ..

Found Drive 0 : HP : C1716T : 3336

For your information, Drive 0 is located at :

=== BOARD:0 BUS:4 TARGET:4 LUN:0 ===

Using compatible configuration file for the drive: eo_worm.tmp]

= = = = =

Loading Drive 1. Please Wait.....

Waiting for drive to be ready ...

Found Drive 1 : HP : C1716T : 3336

For your information, Drive 1 is located at :

=== BOARD:0 BUS:4 TARGET:5 LUN:0 ===

Using compatible configuration file for the drive: eo_worm.tmp]

::

Configuring library unit

Enter physical location for LU qntm_x700_0 : **B1 Lab**

Enter the physical location of the library unit; for example,
"B1 Lab" as shown above.

Selecting the Cleaner Barcode Default

AUTOCONFIG prompts you to accept or decline a default barcode for cleaning cartridges:

```
Would you like to accept the EDM default
barcode of CLNXXX for tape cleaners (y,n)[ Y]? y
```

If you answer **y** (yes), the Library Manager recognizes any tape with a barcode of CLN(000-999) as a cleaner by default and does not place it in the drive during an inventory.

Note: To override this default you must answer **n** (no).

Viewing Log Files

After all library units are configured, lmconfig notifies you that configuration is complete. AUTOCONFIG then asks if you want to view the log files. If you answer yes, information for each library unit appears:

```
Finished configuring library unit(s).
```

```
Would you like to see the log files (y,n)? y
```

```
=====
=      Library Unit Info
= Vendor ID   = HP
= Product ID  = C1710T
= Product Rev = 6.10
=====
( Device name : Board/Bus/Target/LUN : Vendor ID :
Product ID : REV )
Robot 0 : 0 4 0 0 : HP : C1710T : 6.10
Drive 0 : 0 4 4 0 : HP : C1716T : 3336
Drive 1 : 0 4 5 0 : HP : C1716T : 3336

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

Completing lmconfig

After the library managers are configured, type **q** to exit the lmconfig utility.

Main Menu

```
1 LIST          current Library Manager configurations
2 INSTALL      EMC drivers
3 UPDATE       EMC drivers
4 REMOVE       EMC drivers
5 AUTOCONFIG   Automatically configure all library units
6 DECONFIGURE  a Library Manager
7 HELP
```

Choose the configuration operation you want
(1,2,3,4,5,6,7,q)? **q**

Reboot the EDM system by entering the following:

```
# shutdown -y -i6 -g0
```

This important step starts the vmdaemon, and ebfsd and vmdupd daemons. EDM software does not run until these daemons start.

A full inventory begins after the reboot; the time period for completing an inventory depends on the amount of media that the library unit contains.

Deconfiguring a Library Manager

When you deconfigure a Library Manager, lmconfig deletes the Library Manager's subdirectory and its contents. You should deconfigure a Library Manager when you permanently remove a library unit from the server.

To deconfigure a Library Manager, do the following:

1. Start lmconfig and choose 6 DECONFIGURE in the main menu.

2. `lmconfig` lists the currently installed Library Managers; at the prompt, select the one you want to remove:

```
Choose one or more Library Manager configurations to be removed
1 offline_0
2 offsite_0
3 at_452_0
4 hp_c17xx_0
Enter comma-separated choice(s) on a single line (1, 2, 3, 4, q)
```

3. Enter the number(s) that correspond to the Library Manager(s) that you want to deconfigure (as shown in the example above).

Note: If you inadvertently remove the offline or offsite Library Manager, `lmconfig` automatically adds it back for you. Just choose the `CONFIGURE` option in the `lmconfig` main menu.

```
offsite_0 removed
```

```
qntm_x700_0 removed
```

```
hp_c17xx_0 removed
```

```
offline_0 removed
```

The utility removes the Library Manager's subdirectory and its contents from `/usr/epoch/etc/lm`. It also deletes the Library Manager from the `vm.cfg` file and notifies the Volume Manager to reread the file and kill the associated LM daemon.

4. When the main menu appears, select **q** (quit) to return to the system prompt.

Main Menu

1 LIST	current Library Manager configurations
2 INSTALL	EMC drivers
3 UPDATE	EMC drivers
4 REMOVE	EMC drivers
5 CONFIGURE	Manually configure a library unit
6 AUTOCONFIG	Automatically configure all library units
7 DECONFIGURE	a Library Manager
8 HELP	

Choose the configuration operation you want (1,2,3,4,5,6,7,8,q)? c

If You Have Trouble Configuring a Library Unit

If a drive other than drive 0 fails while configuring a library unit, AUTOCONFIG asks whether you want to configure the LU with the drives that AUTOCONFIG was able to find (this number is less than the total number of drives that the LU contains).

Messages that are similar to the following appear:

```
*** ERROR: Cannot find node for drive 1
***      Drive 1 may have been loaded using a
***      cleaning cartridge, or it may be damaged or disabl
***      Configuration of library unit #0 has stopped.
```

```
AUTOCONFIG could only find the first 1 drive(s)
```

```
Would you like to configure the library unit
with only 1 drive(s) (y,n)[ N]? no
```

If you enter **n** (no), configuration of the library unit stops:

```
_autoconfig failed: *** Error: SCSI location of
drive #1 was not found
*** Error: Unable to configure library unit #0
```

If you enter **y** (yes), configuration completes with the drives that AUTOCONFIG found.

NOTE: There is a tape in drive 1

Please shut down EDM server after AUTOCONFIG finishes, and manually remove the tape from the drive.

If a Problem Occurs While Configuring Multiple Library Units

If you want to configure more than one library unit at one time and AUTOCONFIG fails because a volume is stuck in a drive, AUTOCONFIG may have a problem while removing the volume from the drive.

Turn off the library unit, remove the volume from the drive, and check the drive cables. Then restart EDM and restart AUTOCONFIG. This enables AUTOCONFIG to configure the remaining library units.

18 Man Page Listing

This chapter lists the man pages for backup and restore commands, volume management commands, media duplication command, and HSM commands. You can display a full description of each command by typing **man** followed by the command name.

The following three categories of man pages are described:

- Backup and Restore Man Pages
- Volume Management Man Pages
- HSM Man Pages

Backup and Restore Man Pages

This section describes backup and restore commands that can be run from the EDM server. These commands can be used, among other things, to initiate backups (**ebbackup**) and restores (**ebrestore** and **ebcrecover**), start the EDM window (**edm** and **edmrestore**), and generate a variety of reports (**ebreport** and **ebcreport**) directly from the EDM server's command line.

Backup and Restore Commands and Daemons	Description
eb	Introduces the EDM Backup product, programs, daemons, and man pages.
eb_build_hstab	Updates the database of hosts known to the EDM Backup Client Installation wizard.
eb_dc_backup	Starts an EDM Symmetrix Connect backup.
eb_dc_restore	Restores data backed up via a direct connect backup.
eb_deinstall_client	Deinstalls the EDM Backup client software using the command-line interface.
eb_deinstall_server	Deinstalls the EDM Backup server software using the command-line interface.
eb_install_client	Installs the EDM Backup client software (called by the EDM Backup Client Installation Wizard).
eb_install_server	Configures the EDM Backup server software (called by eb_server_config).
eb_rehome_client	Updates an EDM Backup client to know that it has a different EDM Backup server than the one from which it was originally installed.
eb_server_config	Configures a host for use as a backup server or deconfigures it.
eb_sybconf_db	Configures online Sybase database backups.
ebbackup	Initiates and controls client backups under EDM Backup.

Backup and Restore Commands and Daemons	Description (Continued)
ebbbackupd	Performs a backup of a single client system under EDM Backup.
ebcatalogd	Supervises the post-processing of backup catalogs.
ebcatclean	Deletes backups, catalogs, and saveset records that expired or are unreferenced under EDM Backup.
ebcatcomp	Completes the information stored in a backup catalog file and creates backup catalog deltas.
ebcatproc	Forces the processing of a backup catalog.
ebcatsort	Sorts an EDM Backup catalog file.
ebcp	Copies data from one place to another on an EDM Backup server, an EDM Migration client, or between two such machines.
ebcrecover	Provides an easy way to execute the ebrestore command from an EDM Backup client using the client's native connection method. (It actually calls ebrecover on the EDM, which is just a symbolic link to the actual restore program, ebrestore .)
ebcreport	Enables users to run ebreport from an EDM Backup client using the client's native connection method.
ebexpire	Deletes backup data, saveset records, and backup catalogs that have expired under the EDM Backup system.
ebfs_dump_vol	Reads a volume of backup media producing a hexadecimal listing of its contents an/or an extended-cpio stream that can be read by recxcpio to restore a directory hierarchy.
ebimport	Imports backup media, backup catalogs, and saveset records.
eblistend	Listens for requests from DBMS clients for backups and restores, and starts them as needed.
ebrestore	Restores backup files from backups created by the EDM Backup system.
ebreport	Produces several backup reports including media, history, and disaster reports. (See Chapter 16 "Backup Reports and Log Files".)

Backup and Restore Commands and Daemons	Description (Continued)
ebtreegen	Generates a tree index for a backup catalog file. (Evoked by ebcatalogd .)
edm	Starts the EMC Data Manager graphical user interface.
edmcrestore	Starts the EDM Restore window and displays it on the backup client.
edmhlp	Starts the online Help facility for the EDM.
edmlinkq	Allows you to query the remote client via the EDM Transfer Protocol to provide version and capability information. This has the side-effect of allowing the client-server connection to be tested.
edmproc	Lists, starts up, shuts down or restarts all edm daemon processes. edmproc performs the start up/shut down operations in the correct order.
edmremote	Starts the EDM GUI on a remote EDM server and displays it on a specified host.
edmreport	Executes saved reports on currently running or completed backups on a local EDM or on an EDM Domain. If using a domain, the EDM Domain Master machine must be a trusted machine, because the local administrator can change passwords and run the edmreport CLI without knowing the Domain Login Credentials.
edmrestore	Starts the EDM Restore window on the EDM Backup server.
epcleanup	Removes files that are no longer needed. Usually run from crontab.
epcomm_util	Command line utility for communicating with the EDM Client Communications Daemon, epcommd.
epnewlog	Rotates, archives, or truncates system logs. Usually run from crontab.
epshowmod	Displays all or selected installed EDM modules.
epshowpath	Displays the installation location of EDM software.
epshowprod	Displays information for a selected or all installed products.

Backup and Restore Commands and Daemons	Description (Continued)
epshowvers	Displays all products and their versions available on the EDM distribution CD.
eptrunclog	Truncates the daily message log file and mails a copy to specified users. This is usually run from crontab.
findxcpio	EDM Backup's client find and xcpio program.
ntexchreport	Reports on the Microsoft Exchange backups that have been made to an EDM server.
ntexchrestore	Restores a Microsoft Exchange object that was backed up by the EDM server.
ntsqlreport	Reports on Microsoft SQL Server database backups made to the EDM server.
ntsqlrestore	Restores a Microsoft SQL Server object that was backed up by the EDM.
olddb_exec	Initiates an Oracle Online backup or restore from the EDM.
portservices	Modifies the edm_services files on the EDM, enabling, changing, and disabling port control settings.
rasd	RASD (Reliability Agent Scanner daemon) monitors significant system events that inhibit the successful completion of EDM applications.
recxcpio	Creates a directory hierarchy that corresponds to the contents of an extended-cpio stream, such as that produced by findxcpio and the ebrestore program.
snmpeved	The SNMP subagent for Volume Management and EDM Backup.
sybrecover	Recovers striped database back ups from the EDM to a Sybase server.
sybreport	Reports on Sybase backups made to the EDM server.

Volume Management Man Pages

The Volume Management command line interface (CLI) enables you to monitor volume management activities and perform administrative tasks at the command line level. You can use the CLI in a non X windowing environment (such as dial-in-sites). The CLI also enable you to write shell scripts to automate volume management tasks.

Volume Management Commands and Daemons	Description
dbreport	Generates volume reports from the volume database.
edmlm	Starts the graphical EDM Library Unit Manager separate from the EDM GUI..
evmaddtempl	Creates a volume template for a specified media type and application.
evmchtempl	Modifies a volume template for a specified media type and application. This can be used to increase maximum usage of a template.
evmchvol	Changes the attribute settings for an existing labeled volume. This can be used to increase maximum usage of a volume.
evmclean	Cleans drive(s) in a named library unit. Before cleaning a drive, the cleaning cartridge must be present in the library unit. (See Description .)
evmctl	Queries or sets attributes of the EDM Volume Management system or individual library units.
evmeject	Ejects the specified volume or cleaning cartridge from a library unit. This works only with library units that have an inlet. (See Description to get a volumes identifier.)
evmenable	Enables a drive or library unit. Volume management disables a drive when certain errors occur.
evmimport	Imports one or more volumes into a server's volume catalog. Use this command when you move volumes from one server to another or to reconstruct a volume catalog that was destroyed.
evminject	Inserts a volume into a library unit. (See Description .)

Volume Management Commands and Daemons	Description (Continued)
evminventory	Initiates an inventory of the volumes in the named library unit.
evmlabel	Labels the selected volume by using a specified volume allocation template. Use Description -t to list the names and IDs of all defined templates and Description to view a template's attributes.
evmlistd	Lists all running volume management processes.
evmmount	Queues a mount request for the specified volume. Generally, volume management handles all mount and dismount requests. Therefore, this command is intended only for mounting foreign volumes.
evmreject	Rejects a queued request for a volume.
evmrmtmpl	Removes the specified volume template from the volume management template database.
evrmrmvol	Removes all knowledge of an existing volume from a server's volume catalog.
evmseterror	Enables you to set the error and/or warning count for a drive, volume, or library unit.
evmstat	Provides status of volume management system, including devices (library units and drives), volumes, media types, and system notifications.
evmtmpl	Displays attributes of one or more volume templates.
evmumount	Removes a volume from a drive that was mounted by using evmmount .
evmvol	Displays attributes of one or more volumes.
evmwhere	Describes the attributes that are obtained by running evmvol -V . This is NOT a command. This man page describes the where -clause syntax only.
lmconfig	Configures library managers and device drivers for the EDM.

Volume Management Commands and Daemons	Description (Continued)
vmdaemon	EDM volume management daemon. Note: This command should not be run from the CLI.
vmdup	Manages media duplication for specified trails or volumes by turning on duplication for manually-duplicated trails. This is also used to reschedule duplication for a given volume, in case a duplication failed.
vmdupcfg	Displays the current values of the duplication configuration parameters, and allows the values to be changed.
vmdupd	Run from the command line to alter the state of the currently-running vmdupd daemon that controls the media duplication, as well as to display the list of volumes currently scheduled for duplication. Only one occurrence of the vmdupd (run without arguments) can run at a time.

HSM Man Pages

This section describes the following types of migration commands: staging and staging configuration commands, network migration server commands, and user level commands.

Most users do not need the user level commands. These are useful for those who set up application environments and for those who need to understand filesystem usage patterns.

User level commands are marked with an asterisk (*) in the following table.

HSM Commands and Daemons	Description
ebcheck	Finds files that have inconsistencies with their staging IDs and fixes them.
em_new_volume	Allocates a new staging volume for a specific staging template.
embsi *	Stages specified files in from staging media or client stores, which ensures that they are completely resident on magnetic storage.
emcheck *	Checks the HSM client and server configuration to verify its correctness, warns you of potential problems, and corrects inconsistencies.
emchmod *	Sets the staging control properties for a file or directory. Unlike chmod , it clears unspecified properties.
emcompact	Automatically compacts staging volumes. Usually run from crontab.
emcrecover_wait *	Waits for a set of client store bitfiles to be restored on an EDM with HSM. It scans the set of files listed on the command line and checks the status of every bitfile referenced by this set.
emcreport *	Displays information about client store usage and identifies the current staging targets for each stageable filesystem.
emdu *	Displays the number of KB contained in all files and directories specified. Using the -v option displays the amount of virtual space, which includes the space on the EDM if the files have a staging image; otherwise it is the space on the local magnetic disk.

HSM Commands and Daemons	Description (Continued)
emfind *	Recursively descends the directory hierarchy for each pathname in the pathname-list, seeking files that match a logical expression. Supports several additional predicates over find .
emfsconf	Configures HSM filesystems by assigning filesystems to staging templates, removing filesystems from staging templates, and changing filesystem parameters.
emfsdeconfig	Deconfigures a migration filesystem.
emfsreport *	Produces virtual filesystem statistics. It displays the amount of stageable filesystem data, and the amount of data that is currently staged. It displays your working set in days of usage.
emls *	Lists file attributes. Similar to ls , but lists staging attributes, including the number of KB on magnetic disk, the number of KB currently staged out, and server and client store if any, to which the file is staged, as well as the staging control properties, such as residence priority.
emlsconf	Displays the current staging configuration parameters.
emscheck	Checks the network migration server configuration files. The emscheck command checks all global and store specific files for syntactic and semantic correctness. In addition, it performs certain clean up operations on client stores. Run emscheck nightly via cron .
emschs	Changes certain configuration parameters for an existing store.
emsconfig	Changes the protocol interface parameters. Only use at the direction of your customer service organization.
emsd	The network migration server daemon. The emsstart command always invokes this daemon during system startup. The daemon handles all network client HSM protocol requests from the client systems.
emsdefs	Changes certain default store configuration parameters. The emsdefs command affects the operation of the emsmks command.
emshalt	Terminates the network migration server. The emshalt command kills the running emsd processes and aborts any outstanding staging operations.

HSM Commands and Daemons	Description (Continued)
emsinit	Initializes the network migration server configuration files. If no configuration files exist, emsinit creates an initial configuration with no stores and standard default definitions. The emsinit command is normally only run during installation.
emslss	Lists all configured stores, or individual stores selected by name or owning client.
emsmks	Makes a client store. You can also use emsmks to insert an existing store tree into a server configuration database.
emsmvs	Renames and/or moves an existing store.
emsrms	Removes an existing store.
emsstart	Initiates the network migration server. This has no effect if the network migration server is already running. The emsstart command is run during system startup.
emsstat	Displays network migration server usage statistics. It displays both cumulative and incremental statistics. Use emsstat to determine the current status of network migration services.
emstage *	Explicitly stages out the specified files. You must be the file owner or the superuser to use this command.
emstconf	Creates new staging templates, removes existing staging templates, and changes parameters for existing staging templates.
emsundel	Starts a restore run to retrieve bitfiles listed in the recover_list files. Only run emsundel when an operator is available to handle volume mount requests.
emsysconf	Sets system-wide staging parameters in the configuration database.
emvck	Checks and corrects staging volume statistics. This is usually run from crontab.
restage	Stages or restages files to the specified staging trail.

Part V
Disaster
Recovery

19 Being Prepared for a System Disaster

If there is a disaster, you should be prepared for a disaster recovery. However, feel free to call Customer Service with questions.

CAUTION: Do not wait for a disaster to read this chapter. The information in this chapter is about steps that must be taken with each backup so that you will be able to recover when a disaster occurs.

This chapter tells you how to prepare for the disaster recovery of your backup files. The following two chapters contain overall disaster recovery procedures to use when you experience a disk crash on an EDM server or client. Because each disaster is unique, these steps are presented only as a guide and not as all-inclusive, step-by-step instructions.

CAUTION: Performing a disaster recovery requires experience with EDM Backup administration (and HSM administration, if you have the HSM option), UNIX system administration, and the site environment.

You must establish a disaster strategy and safeguard your media and the reports *before* a disaster occurs. Otherwise, you will not have the necessary information to recover your system to its original state.

Note: You should develop a disaster recovery plan that meets your specific organizational goals. Actual file recovery can be as simple as off-site tapes, or as complex as duplicate Symmetrix systems utilizing SRDF/Timefinder.

To fully recover from a system disaster, you must run regular backups, safeguard your backup media, and run and save the appropriate backup reports. You can provide additional protection by creating redundant media and storing it offsite.

Safeguarding Your Backup Media

To be prepared for a system disaster, you must run regular backups and save the backup media for both the current and previous full rotation periods. For example, assume the following:

- The rotation period is 7 days.
- A full backup is run on Monday.
- Every backup generates a single new piece of media.

If today is Tuesday, you need the media that was generated since Monday of the previous week, or the last eight pieces of media. Tomorrow, you will need the same eight pieces, plus the media that is generated from today's backup.

Be sure to save the backup media in a safe place, preferably offsite or onsite in a fireproof vault.

CAUTION: Failure to have visually identifiable labels on removable media will significantly complicate and lengthen the Disaster Recovery procedure.

Be sure removable backup media is physically labeled (such as with barcodes) so it can be visually identified if needed during the disaster recovery process. If barcoding is not used, each piece of media must be physically labeled with its assigned sequence number.

Running and Saving Reports

At the end of your backups for the day, your LOCAL_DATABASE is automatically backed up, which provides you an exact picture of the EDM Backup database.

At the completion of every LOCAL_DATABASE backup, the /usr/epoch/EB/config/local_db_cleanup script automatically generates a MINIMAL Disaster Report. By default, this report is emailed to all EDM Backup administrators, appended to /usr/epoch/EB/config/disaster-report.log, and printed to the default system printer.

CAUTION: It is essential that you save a hard or soft copy of the MINIMAL Disaster Report after each backup. Keep it in a fireproof location, either offsite or in an onsite fireproof vault.

If the LOCAL_DATABASE work item remains in the schedule for more than 24 hours without being run, it will be forced to run immediately. This is known as a “late” LOCAL_DATABASE backup. The work item remains in the schedule to be run again normally, or forced if needed.

ebbackup displays a message and **ebreport disaster** notes a “late” LOCAL_DATABASE backup.

MINIMAL Disaster Report

This MINIMAL Disaster Report is a subset of the FULL Disaster Report that **ebreport disaster generates**. It provides essential information that you need to perform a disaster recovery on the server—a list of media volumes for the most recent LOCAL_DATABASE backup, the current EDM Backup configuration, the current Library Manager configuration, copies of the key configuration-file settings, and information about baseline backups.

Note: This MINIMAL Disaster Report does *not* include backup client information.

FULL Disaster Report

You should run the FULL Disaster Report once every backup rotation and whenever significant system changes are made. The following example runs the FULL Disaster Report and redirects it to a file:

```
emc# ebreport disaster >  
~sysadmin/disreports/960917
```

See “Backup Disaster Reports” on page 16-19 for a description of the FULL Disaster Report.

Redundant Backup Coverage

In preparation for a possible disaster, it is recommended to have redundant backup coverage so that you can move some backup media offsite for safe keeping. Following are two ways of providing redundant backups.

Configure Alternate Media Sets (Trailsets)

One good backup strategy is to configure an alternate media set (trailset) for use on alternate nights. (A *trailset* contains all of the media that is used in performing full and incremental backups for a backup schedule template in a single rotation period.)

For example, with a rotation period of seven days:

- with a primary trailset only, a complete trailset includes at least one full backup and six incrementals for each work item.
- with an alternate trailset, each complete trailset includes at least one full backup for each work item, but only two or three incrementals.

With primary trailsets only, move backup media offsite as soon as it is older than one rotation period. With alternate trailsets, you can move each backup volume from one trailset offsite as soon as the volume is full.

Media Duplication

Another option is to use Media Duplication, which enables you to create a duplicate set of backup media automatically after each backup session.

After you configure media duplication in the EDM Backup Configuration window, the duplication of a set of backup media occurs automatically after each backup session. You can then take the duplicate media offsite for safekeeping.

This method does not use network bandwidth and can be a good choice if you have extra drives. For more information, see Chapter 9 “Media Duplication”.

20 Recovering a Server from a Disk Failure

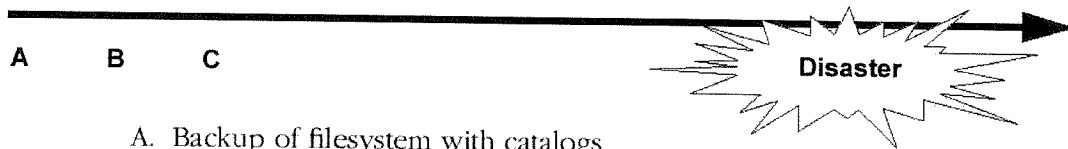
This chapter contains disaster recovery procedures to use when you experience a disk crash on a backup server. Because each disaster is unique, these steps are presented only as a guide and not as all-inclusive, step-by-step instructions.

CAUTION: Performing a disaster recovery requires experience with EDM Backup and HSM (optional) administration, UNIX system administration, and the site environment.

To restore a backup server, determine the extent of the damage and disable backups, replace any damaged disks, reinstall the operating system, and reinstall any lost EDM Backup or HSM software.

Prior to restoring lost files, you import the volume management and backup catalog database information and then use EDM Backup to restore the databases as they were at the time of the last LOCAL_DATABASE backup. After doing this, you can restore all of the data from changes that occurred after the last LOCAL_DATABASE backup.

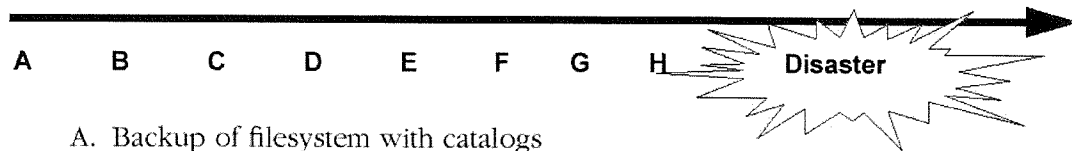
The procedures in this chapter are based upon the example in Figure 20-1 that shows the actions that occurred before the disaster in the simplest case. Figure 20-2 illustrates these and other actions that may have occurred before the disaster.

Figure 20-1**Disaster Immediately after the LOCAL_DATABASE Backup**

- A. Backup of filesystem with catalogs
- B. Remote client work item backup
- C. LOCAL_DATABASE backup

Your disaster could occur immediately after your last LOCAL_DATABASE backup (Step C, above). If so, follow all of the sections except those that are marked "For changes after the last LOCAL_DATABASE backup."

Figure 20-2

Disaster after Other Possible Actions

- A. Backup of filesystem with catalogs
- B. Remote client work item backup
- C. LOCAL_DATABASE backup
- D. Appended backup
- E. Completion of duplication of volumes
- F. Change in configuration of Library Units
- G. ebexpire (run either by cron or manual)
- H. Backups that completed after LOCAL_DATABASE backup

Events, such as D, E, F, G and H in Figure 20-2, may have occurred (in any order) after the last LOCAL_DATABASE backup. If your disaster occurs after one or more of these, read ALL sections in the chapter, including those that are marked "For changes after the last LOCAL_DATABASE backup," and follow those that fit your situation.

Steps to Restore a Server

The actions that are required to restore a server from a disk failure are discussed in this chapter. Follow the steps below to restore a backup server in case of a disk crash or major loss of files. Your situation may vary from this example and you may not need all of these sections. If some filesystems remain intact, you may be able to skip some steps. You should adjust your disaster recovery steps accordingly.

You need information from the Disaster Report for many of these steps. For information about this report, see “Running and Saving Reports” on page 19-3.

Note: The procedures that follow use *emc* as the name of the server. Substitute the name of your own server in each of the examples.

Each of the following steps are described in the following sections of this chapter. Perform them in the order given.

1. Stop All Activity on the Server
2. Reinstall Hardware and Software as Needed
3. Temporarily Reconfigure the Server
4. Restore LOCAL_DATABASE Files

The next two steps are only done if you made changes after the last LOCAL_DATABASE backup.

5. Reconfigure Library Units
6. Restore Catalogs and Backup Information Created After LOCAL_DATABASE Backup

Follow these steps for all disaster recoveries.

7. Restore Data Created Before LOCAL_DATABASE Backup
8. Reenable crontab Entries
9. Restore Past Catalogs
10. Restore Missing Catalogs

Stop All Activity on the Server

Because you perform all of the restore in multiuser mode, you must notify the user community to stop all activity on the server. No one should be able to log in. The server must be inactive before you perform any disaster recovery procedures.

Disable Activity

Edit root's crontab file (/var/spool/cron/crontabs/root) on the server to comment out all backup and HSM staging commands that may start up automatically. Following are commands to comment out:

```
00 0 * * * /bin/kill -1 `cat /usr/epoch/etc/mal/emmasterd.pid` >/dev/null
2>&1#EPCmalc

15 23 * * * /usr/epoch/bin/emvck >/dev/null 2>&1#EPCmalc

00 1 * * * /usr/epoch/bin/emcompact -c >/dev/null 2>&1#EPCmalc

00 2 * * * /usr/epoch/bin/emscheck >/dev/null 2>&1#EPCmalc

00 3 * * * /usr/epoch/bin/emsundel >/dev/null 2>&1#EPCmalc

00 18 * * * /usr/epoch/EB/bin/ebbackup default >/dev/null 2>&1 #EPCebs

00 11 * * * /usr/epoch/EB/bin/ebexpire -expire -purge >/dev/null 2>&1 #EPCebs

00 1 * * * /usr/epoch/EB/bin/ebcatclean -fix_saveset >/dev/null 2>&1 #EPCebs

00 3 * * * /usr/epoch/EB/config/local_db_warning >/dev/null 2>&1 #EPCebs
```

Reinstall Hardware and Software as Needed

Before you perform a restore, you must have your hardware and software in the same condition as that before the disaster occurred.

Determine the extent of the damage to both hardware and software. Be sure to identify and replace all damaged hardware *before* restoring any software.

1. Check hardware and replace if necessary. You can use the disk diagnostics capability of a program such as **format** to determine whether anything is wrong with your disk.

Make sure the replacement hardware is fully compatible with the system you had before the disaster. Each new disk should have at least the same storage capacity as the old disk.

CAUTION: Do not continue with software recovery until you are absolutely certain that the disks are free of hardware problems.

2. Check the filesystems for loss of software, starting with the operating system.

If `/` or `/usr` was destroyed, reinstall the server's native operating system following the platform-specific instructions that are provided with your server. This partitions your disks, restores the root filesystem, and restores the portion of `/usr` that the operating system loads.

To partition the disks correctly, use the disk configuration data from the last Disaster Report (which is described in "Running and Saving Reports" on page 19-3) See Figure 20-3 on page 20-7. Make sure you rebuild these filesystems to the same size (or larger) and the same number of inodes (or more) as they were before the disaster occurred.

Figure 20-3

Locally Mounted Disks from the Disaster Report

Displaying locally mounted disks...

```

/ (/dev/dsk/c0t3d0s0 ):      8192 block size   1024 frag size
230302 total blocks  92988 free blocks  69968 available  60416 total file
56761 free files      8388632 filesys id
   ufs fstype        0x00000004 flag              255 filename length

/usr (/dev/dsk/c0t3d0s3 ):    8192 block size   1024 frag size
673742 total blocks  295032 free blocks  227672 available  169920 total f
153786 free files      8388635 filesys id
   ufs fstype        0x00000004 flag              255 filename length
. . .
. . .
/ep_usr (/dev/dsk/c0t3d0s4 ):  8192 block size   1024 frag size
774766 total blocks  396532 free blocks  319072 available  196352 total f
193845 free files      8388636 filesys id
   ufs fstype        0x00000004 flag              255 filename length
. . .
. . .

```

3. If you have an HSM system, you need to reinstall VERITAS VxFS. See your *VERITAS File System (VxFS) Quick Start Guide*.

Note: If you are reinstalling the operating system, you need a VxFS license.

4. Verify whether the filesystem that contains the backup software is damaged or missing (the EDM Software Filesystem, /ep_usr in the example in Figure 20-4).
5. Check the Disaster Report to see what you had at the time of the last report.

Figure 20-4**Displaying /etc/vfstab... from the Disaster Report**

Displaying /etc/vfstab...

Oct 3 14:23 1998 /etc/vfstab Page 1

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
#						
fd -	/dev/fd fd	- no	-			
/proc -	/proc	proc - no	-			
/dev/dsk/c0t3d0s1	- -	swap -		no	-	
/dev/dsk/c0t3d0s0	/dev/rdisk/c0t3d0s0			ufs	1	no -
/dev/dsk/c0t3d0s3	/dev/rdisk/c0t3d0s3	/usr		ufs	1	no -
/dev/dsk/c0t3d0s4	/dev/rdisk/c0t3d0s4	/ep_usr		ufs	2	yes -
/dev/dsk/c0t1d0s2	/dev/rdisk/c0t1d0s2	/data		vxfs	2	yes -
/dev/dsk/c0t5d0s0	/dev/rdisk/c0t5d0s0	/data1		vxfs	2	yes -
/dev/dsk/c0t5d0s1	/dev/rdisk/c0t5d0s1	/data2		vxfs	2	yes -
/dev/dsk/c0t5d0s3	/dev/rdisk/c0t5d0s3	/data5		vxfs	2	yes -
/dev/dsk/c0t2d0s0	/dev/rdisk/c0t2d0s0	/data6		vxfs	2	yes -
/dev/dsk/c0t2d0s1	/dev/rdisk/c0t2d0s1	/data7		vxfs	2	yes -
/dev/dsk/c0t2d0s3	/dev/rdisk/c0t2d0s3	/data8		vxfs	2	yes -
/dev/dsk/c0t0d0s0	/dev/rdisk/c0t0d0s0	/data3		vxfs	2	yes -
/dev/dsk/c0t0d0s1	/dev/rdisk/c0t0d0s1	/data4		vxfs	2	yes -
swap -	/tmp	tmpfs -		yes	-	

6. Use the Installation Report portion of the Disaster Report to determine which of the various binaries were actually in /usr and which were symbolically linked to another filesystem. The EpochBackup Installation Report section shows the directories and symbolic links for the backup software.

In the third line of the split-install example shown in Figure 20-5 on page 20-9, the backup software is installed in /ep_usr/epoch and a symbolic link is in /usr/epoch and points to /ep_usr/epoch.

Also, in this example, catalogs, db, and log are installed in /data/epoch/EB/ and a symbolic link is in /usr/epoch/EB and points to them.

Figure 20-5**Installation Report in the Disaster Report**

```
EpochBackup Installation Report for server adam at Oct 13 09:01:14 1998
Report options: -all

Installed Software:

EDM High-Performance Centralized Backup with HSM
  abbreviation:  edmhsm
  version:       4.0.0.5
  server platform: sun4_5.5.1
  platform:      sun4_5.5.1
  patch id:      none
  install date:  19971110165742
  modules:       EPCgl EPCtps EPCsnmp EPCesl EPCdevlib EPCdev EPCelmlib EPCelm
  EPCEbhs1b EPChsesdm EPCmalib EPCmalc EPCmawrp EPCEbsedm EPCEbc EPCgui
  EPColdoc

  status:        complete

                    (Installation Report continued on next page.)
```

(Installation Report Continued from previous page.)

Oracle Platinum On-Line Database Backup

abbreviation: oraplt
 version: 1.1.0.5
 server platform: sun4_5.5.1
 platform: sun4_5.5.1
 patch id: none
 install date: 19971112104243
 modules: EPCoraplt
 status: complete

EpochBackup currently running load 7.0.0.0

/usr/epoch IS A SYMLINK to /ep_usr/epoch
 /usr/epoch/EB is a real directory under /ep_usr/epoch
 /usr/epoch/GENDIR IS A SYMLINK to /home/epoch

**Backup
Software**

/usr/epoch/EB/adam is a real directory under /usr/epoch/EB
 /usr/epoch/EB/bin is a real directory under /usr/epoch/EB
 /usr/epoch/EB/catalogs IS A SYMLINK to /home/epoch/EB/catalogs
 /usr/epoch/EB/client is a real directory under /usr/epoch/EB
 /usr/epoch/EB/config is a real directory under /usr/epoch/EB
 /usr/epoch/EB/db is a real directory under /usr/epoch/EB
 /usr/epoch/EB/locks is a real directory under /usr/epoch/EB
 /usr/epoch/EB/log IS A SYMLINK to /home/epoch/EB/log
 /usr/epoch/EB/preconfig is a real directory under /usr/epoch/EB
 /usr/epoch/EB/tmp is a real directory under /usr/epoch/EB

**Backup
Software
Directories**

The local client is of the type: sun_sun4_v55_srv

The client backup username is: ebadmin

The user ID for ebadmin is: 24375

The group ID for ebadmin is: 25

The home directory for ebadmin is: /usr/epoch/EB

Client chip is of type ibm_rs6000_v325 (6.0.0.0), installed Tue May 20 14:05:23 199
 Client yyz is of type windows_nt_all (6.0.0.0), installed Thu Sep 11 15:04:42 199
 Client perf-prol is of type windows_nt_all (5.0.1.0), installed Thu Sep 25 15:58 199
 Client pilgrim is of type sun_sun4_v54 (6.0.0.0), installed Thu Oct 23 15:53:58 199
 Client adam is of type sun_sun4_v55_srv (6.0.0.0), installed Wed Nov 12 11:03:07 199

End of EpochBackup Installation Report for server adam at Mar 13 09:01:14 1998

7. Reinstall the backup software.

Use the data from the Installation Report section of the Disaster Report to determine where and how it had been installed most recently (see Figure 20-5 on page 20-9).

- a. If the filesystem that contained the backup software was destroyed, reinstall the software as a scratch installation. See your *Software Installation* manual.
- b. If the filesystem that contained the backup software is still intact but /usr was destroyed, you need to recreate the symbolic link from /usr that points to this filesystem, which contains the backup software.

To match the example in Figure 20-5 on page 20-9, enter the following (specify your own installation directory):

```
emc# ln -s /ep_usr/epoch /usr/epoch
```

Temporarily Reconfigure the Server

You must configure the library unit by using **lmconfig**, and then a temporary backup configuration by using **eb_server_config**.

lmconfig

1. Obtain the **lmconfig** data from the Disaster Report (see Figure 20-6).

Figure 20-6**Library Manager Configuration from the Disaster Report**

Displaying library manager configuration (used with lmconfig...)

* Lu_name	Name	ID	Status
L offline_0	-	-	synced
L offsite_0	-	-	synced
L at_dlt_3264_0	-	(0,1,1,0)	synced
D at_dlt_3264_0	drive_0	(0,1,5,0)	enabled
D at_dlt_3264_0	drive_1	(0,1,4,0)	enabled
D at_dlt_3264_0	drive_2	(0,1,3,0)	enabled
L hp_mf_c17xx_0	-	(0,2,6,0)	synced
D hp_mf_c17xx_0	drive_0	(0,2,5,0)	enabled
D hp_mf_c17xx_0	drive_1	(0,2,4,0)	enabled
D hp_mf_c17xx_0	drive_2	(0,2,2,0)	enabled
D hp_mf_c17xx_0	drive_3	(0,2,1,0)	enabled

2. Remove all of the media from the library unit that you just configured and put the media aside, in their own box or some other location where they cannot be confused with other media. You reuse these media when you restore the catalogs (see page 20-23).

This significantly shortens the time that is required to inventory this library unit when you reboot the server.

3. Using **lmconfig**, install the drivers for a library unit that supports the media you use later to import and restore the LOCAL_DATABASE backup.

See Chapter 17 “Configuring Library Managers” for details about running **lmconfig**.

Note: You can save a significant time by configuring only one library unit, even if more than one is available.

4. Use the **CONFIG** option of **lmconfig** to configure the library unit.

Note: If you use the **AUTOCONFIG** option, you must leave at least one piece of media in the TLU. If you do this, leave the piece of media required for LOCAL_DATABASE restore.

5. Using the LOCAL_DATABASE volumes section of the Disaster Report (see Figure 20-7 on page 20-16), reinsert the volumes that contains the LOCAL_DATABASE backup into the library unit that you configured.
6. Reboot the server again with the following command:
`emc# /usr/sbin/shutdown -y -i6 -g0`

The Library Manager performs an inventory of this library unit. When the inventory is complete, continue to the next step.

eb_server_config

Configure a temporary EDM Backup configuration. The original EDM Backup configuration is restored later, when you restore the LOCAL_DATABASE files as described in the next section.

1. Determine whether the backup server software was a split directory or single directory server configuration.

Look at the Installation Report in the Disaster Report (see Figure 20-5 on page 20-9).

- a. If catalogs, db, and log are real directories, not SYMLINKs, you have a single directory installation.

- b. If either catalogs, db, or log are SYMLINKs to another directory, you have a split directory installation.
- 2. Run **eb_server_config** and respond to the prompts to recreate the installation and configuration of the server, as indicated in the Installation Report.
 - a. If the previous installation was a single directory installation, you need to select *no* when asked if you wish to install into a “split” directory.
 - b. If the previous installation was a split directory installation, you need to select *yes* when asked if you wish to install into a “split” directory.
 - c. Determine the parent directory of the catalog, db, and log directories (in Figure 20-5 on page 20-9) that would be /usr/epoch/EB, and enter that path when asked for the name of the target directory.
 - d. Determine the client backup username by examining the Installation Report of the Disaster Report (see Figure 20-5 on page 20-9), and enter it when asked.
 - e. Answer the remaining questions as you did in the original installation and as shown in the Disaster Report.

Restore LOCAL_DATABASE Files

Once all hardware and software are in place, and you temporarily reconfigured the server, you need to restore your most recent LOCAL_DATABASE backup.

1. Import the original or the duplicate volumes that contain the LOCAL_DATABASE backup (which was inserted at step 5 on page 20-13) into the Volume Manager catalog. To do this, enter:

```
emc# evmimport -l LibraryUnitName -a
```

where *LibraryUnitName* is the name of the library unit you configured earlier in the procedure. The **-a** option causes all of the volumes in that library unit to be imported.

2. Use the LOCAL_DATABASE section of the Disaster Report (see Figure 20-7) to get the saveset ID and first volume ID of the LOCAL_DATABASE backup.

Note: If you plan to use duplicate volumes, you only need to load the duplicate volumes in the library unit. If the original volume is in the offline state, the substitution of duplicates happens automatically. Always use the original volume ID when needed in the CLI commands or in the ebimport CLI, do not use the duplicate volume ID.

Figure 20-7

LOCAL_DATABASE Section of the Disaster Report

Saveset ID	Original Volume ID	Duplicate Volume ID
EDM Backup MINIMAL Disaster Report for server "elf" on May 18 13:05:42 1999		
LOCAL_DATABASE Backup Volumes Report		
The following volumes contain the most recent LOCAL_DATABASE backup which will be required in the event of a Disaster Recovery:		
Saveset ID 7271A0D4.37408006 for LOCAL_DATABASE backup on 5/17/99 16:45		
	backup_DLT #0010 (38DD01FC8C977D94) [original] - currently in library unit "de_dlt_x700_0", slot #5	
	backup_DLT #0011 (EBDD020907E79820) [duplicate] - currently in library unit "de_dlt_x700_0", slot #3	
This LOCAL_DATABASE backup will require 27.2 MB of disk space to be recovered		

3. Import the LOCAL_DATABASE backup to restore the saveset records and catalogs.

Run **ebimport** using the **-media** option (with the 16-character original volume ID listed in parentheses for the LOCAL_DATABASE backup volume) and the 17-character LOCAL_DATABASE saveset ID, as follows:

```
emc# ebimport -media 38DD01FC8C977D94 7271A0D4.37408006
                        |                      |
                    Original Volume ID      Saveset ID
```

At this point, enough data is restored to restore the LOCAL_DATABASE backup.

4. Make sure catalog processing completed. To do this, enter:
emc# **ebcatalogd -status**

When it reports that all catalogs are processed, continue to the next step.

5. If you have an HSM system, reconfigure all migration controlled filesystems that must be restored.

Refer to **emlsconf** output (see Figure 20-8 on page 20-18) in the Disaster Report and use exactly the staging parameters that it reports as arguments to the **emsysconf**, **emstconf**, and **emfsconf** commands. Valuable control information was lost when the filesystems were damaged then recreated. This control information is recreated when you run **emfsconf**.

Figure 20-8**EpochMigration Local Configuration from the Disaster Report**

```

EpochMigration Local Configuration

EpochMigration System Configuration:
  Enable_stage_outMax_trailsEnable_self_describing
Y           3           N

Staging trail "Retrieve_random"
  Stage outs enabled: Y Media: EO      Unrestricted
  Self-Describing enabled: N
  Enable   HWM   LWM   PSWM   Delay   Mntpoint
Y       68    34    17       0 defaults for Retrieve_random

Staging trail "Retrieve_cached"
  Stage outs enabled: Y Media: EO      Unrestricted
  Self-Describing enabled: N
  Enable   HWM   LWM   PSWM   Delay   Mntpoint
Y       95    88    80       0 defaults for Retrieve_cached

Staging trail "Archive"
  Stage outs enabled: Y Media: EO      Unrestricted
  Self-Describing enabled: N
  Enable   HWM   LWM   PSWM   Delay   Mntpoint
Y       68    34    17       0 defaults for Archive

Staging trail "Trail_1"
  Stage outs enabled: Y Media: EO      Unrestricted
  Self-Describing enabled: N
  Enable   HWM   LWM   PSWM   Delay   Mntpoint
Y       94    78    64       0 defaults for Trail_1
Y       d     d     d       d /data
Y       95    88    80      20 /data2
Y       95    88    80      30 /data3
Y       90    68    45      70 /data7
Y       95    88    80      10 /data1
Y       95    88    80      60 /data6
Y       95    88    80      40 /data4
Y       95    88    80      50 /data5

```

6. Select a temporary location for LOCAL_DATABASE.

CAUTION: You cannot use a filesystem that is enabled for HSM for the temporary location of the LOCAL_DATABASE. (See Figure 20-8 on page 20-18 for the configuration of such a filesystem.)

CAUTION: The LOCAL_DATABASE backup must *not* be restored in place (to its original location), or the temporary database files that were just created and are being used to restore the original database are overwritten. This causes the disaster recovery process to fail.

You must have a filesystem with enough free disk space to restore the LOCAL_DATABASE backup to a temporary location. The amount of required disk space is noted in the LOCAL_DATABASE portion of the Disaster Report (see Figure 20-7 on page 20-16). If the filesystem you choose is the same filesystem that contains the EDM software, you need twice as much free disk space as noted in the Disaster Report.

7. Use **ebrestore** (*not* the EDM Restore window) to restore the LOCAL_DATABASE backup to the temporary location determined above, such as
/newpath/newdir/temp_local_db.

This restores critical files originally located in /usr/epoch.

```
emc# mkdir -p /newpath/newdir/temp_local_db
emc# ebrestore -w emc:LOCAL_DATABASE -c emc -D emc
-d /newpath/newdir/temp_local_db /
```

Note: In “-w emc:LOCAL_DATABASE,” “-c emc,” and “-D emc,” “emc” is the name of the server.

8. Now run a full label and barcode inventory to avoid duplicate entries in the catalog. You can run an inventory by using the **evminventory** command (refer to the **evminventory** man page), or through the Library Unit Manager window. (Refer to EDM online help for more information.)

9. Stop the following on the server as shown:

- a. Stop catalog processing by halting the **ebcatalogd** daemon. To do this, enter:

```
emc# ebcatalogd -halt
```

Note: **emfmd** must be running on HSM servers.

- b. List the applicable backup, volume management, and HSM daemons with the following command:

```
emc# edmproc
```

- c. Stop the daemons with the following command:

```
emc# edmproc -shutdown
```

- d. Verify that all processes are stopped.

```
emc# edmproc
```

On an HSM system, **emfmd** should still be running. On a system without HSM there should not be any daemons still running.

10. After the required shutdowns, move the LOCAL_DATABASE from the temporary location to its permanent location.

Note: There are two copies of **eb_disaster_move**. Be sure to use the copy relative to the temporary location of LOCAL_DATABASE as shown in these instructions.

CAUTION: Do NOT use the copy in:
/usr/epoch/EB/config/

To do this, change to the temporary location:

```
emc# cd /newpath/newdir/temp_local_db
```

Then find the path of the relative **eb_disaster_move**, cd to it and verify that you are there:

```
emc# find . -name eb_disaster_move -print
```

```
emc# cd <path containing eb_disaster_move>
```

```
emc# pwd
```

Move the LOCAL_DATABASE to its permanent location by using **eb_disaster_move**:

```
emc# ./eb_disaster_move
```

Note: Regarding use of duplicate volumes:

When `eb_disaster_move` completes, it automatically checks for failed and uncompleted duplications. If a duplication failed, you must use the original volume for restore. If the duplication had not completed or was scheduled, at the time the catalog was written to tape, the duplicate will be available for use.

If the disaster occurred soon after the backup completed, these duplications may not have completed. If there are errors reported during the restore while using the duplicate, use the original volume. If you do not have the original, contact Customer Service to restore from an earlier duplicate.

After you have completed the disaster recovery, reschedule the duplication of those original volumes that were not successfully duplicated.

`/newpath/newdir/temp_local_db` remains after **`eb_disaster_move`** is done but is no longer needed. You may wish to delete it after completing all disaster recovery procedures.

Any modifications to the `eb.cfg` file that were made after the LOCAL_DATABASE backup must be added manually to the restored `eb.cfg` file. You cannot restore any of the catalogs after the `local_db` backup (such as a new work item), until this is done.

Note: If you want to use duplicate volumes that were made after the LOCAL_DATABASE backup, follow the instructions in "Restore Catalogs and Backup Information Created After LOCAL_DATABASE Backup" on page 20-23.

11. Restart the following as shown:

a. Restart the Volume Management daemon with the following command:

```
emc# sh /usr/epoch/etc/rcS/S20elm start
```

b. Restart **ebfs** with the following command:

```
emc# sh /usr/epoch/etc/rcS/S30ebfs start
```

c. If you have an HSM system, reenable staging with the following command:

```
emc# sh /usr/epoch/etc/rcS/S40mal start
```

You do not restart the **ebcatalogd** daemon at this time. It is restarted after you reboot the server. See “Stop Backups” on page 20-23.

12. Eliminate any incomplete backups or catalogs that could confuse catalog processing. To do this, enter:

```
emc# ebexpire -partial -purge -expire
```

13. Ensure that all catalogs that you just restored are in sync with the saveset database. To do this, enter:

```
emc# ebcatalogclean -disaster
```

This completes the restore of the LOCAL_DATABASE backup which contains the server catalogs and backup information at the time of the last LOCAL_DATABASE backup. The LOCAL_DATABASE is no longer available (or needed) for the restore process. If you need it again, start at step 6 on page 20-19.

At this point, the server's library units are configured as they were at the time of the last LOCAL_DATABASE backup. If there were no significant events after the LOCAL_DATABASE backup (Step C in Figure 20-1 on page 20-2), enter **ebcatalogd** on the command line to restart the daemon and skip to “Restore Data Created Before LOCAL_DATABASE Backup” on page 20-28.

Reconfigure Library Units

For changes after the last LOCAL_DATABASE backup

If any library units changed their configuration *after* the LOCAL_DATABASE backup was generated (step C in Figure 20-2 on page 20-3), use **lmconfig** to reconfigure them. There is no information in the Disaster Report to assist you in this step, since these changes (if they occurred) happened after the Disaster Report was generated.

Restore Catalogs and Backup Information Created After LOCAL_DATABASE Backup

For changes after the last LOCAL_DATABASE backup

If any of the events, such as D, E, F, or G in Figure 20-2 on page 20-3, occurred after your last LOCAL_DATABASE backup, you should get the catalogs and backup information back as soon as possible.

This applies, if you are using duplicate media which completed after the LOCAL_DATABASE backup.

To do this, you need to stop all backups, run a full inventory, import volumes, and then restore the catalogs and backup information.

Stop Backups

Disable backups by commenting out any **ebbackup**, **ebexpire**, and **ebcatclean** commands in the root crontab file. This needs to be done again because root crontab may have been modified when you moved LOCAL_DATABASE from the temporary location to the permanent location.

Reboot the server with the following command:

```
emc# /usr/sbin/shutdown -y -i6 -g0
```

This starts **vmdaemon** and **ebcatalogd** and makes all of your library units available to use in the remaining restore process.

Run a Full Inventory and Import Uncataloged Volumes

1. Take the box of tapes you set aside in step 2 on page 20-12 and insert the tapes in the library unit.

2. Inventory the contents of the library unit:

```
emc# evminventory -l at_dlt_452_0 -a -L
```

Where -l = library name, -a = all slots, and -L = Verify Label

3. Then import all uncataloged volumes in the library unit.

```
emc# evmimport -l at_dlt_452_0 -a
```

Where -l = library_name, -a = all slots (uncataloged only)

4. Wait until the inventories complete before proceeding.

Import Backup Catalogs and List Volumes

Next import the current catalogs for the current rotation.

1. To identify these, run **ebreport media** and select the most recent volume in the most recent rotation for each trail (as noted by a "*" in the example in Figure 20-9).

```
emc# ebreport media
```

Figure 20-9**Output from ebreport media**

EDM Backup Media Report for server missile on May 9 14:22:22 1999

Report options: none

Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary Trailset

09/30/1998 12:54:42 Rotation ID:4CD84987.F6BECF8D.00000200.54028F30, 4 backups

Media duplication used on 1 copy

*Orig Vol: 60D84A1170094B3E (BNY574), Seq #: 000024 in TLU: at_dlt_3264_0, media: DLT

Dup Vol: 73D8745B3E0384A5 (BDE133), Seq #: 000028 in TLU: at_dlt_3264_0, media: DLT

Duplication State: Done, Successful, Duplication Date 05/08/1999 16:06:04

Orig Vol: 4CD84987F6BECF8D (ASV891), Seq #: 000027 in TLU: at_dlt_452_0, media: DLT

Dup Vol: 96D8746209A96A98 (BDE128), Seq #: 000031 in TLU: at_dlt_452_0, media: DLT

Duplication State: Done, Successful, Duplication Date 05/08/1999 16:25:04

.
.
.

EDM Baseline Media Report for server adam on May 9 13:10:26 1999

2. Run **ebimport** to import the backup catalogs and information from the volumes that were current at the time of the last successful LOCAL_DATABASE backup (identified by an * in the output from **ebreport media**, as shown in the above example).

These volumes may also contain appended backups not known at the time of the last LOCAL_DATABASE backup. (These volumes may reside inside or outside the library units.)

Note: This portion of the restore can take a considerable amount of time because EDM Backup has no record of the backups or the volumes on which they reside.

3. For each volume, run the following command to import catalogs and backup information created after the time of the LOCAL_DATABASE backup:

```
emc# ebimport -media valid -clever 9 -level 9
```

Some catalog imports may fail. If so, stop the catalog daemon and restart it with the **30 second** option, which causes these catalogs to be processed again in 30 seconds rather than the default value of one day.

```
emc# ebcatalogd -halt
```

```
emc# ebcatalogd -retry_time 30
```

If you make this change you should reset **ebcatalogd** after completion of the disaster recovery. To do this, enter:

```
emc# ebcatalogd -halt
```

```
emc# ebcatalogd -catalogs 3 >& /dev/null
```

4. Execute the following command to list all volumes in the library unit and their states:

```
emc# evmstat -v
```

If there are any “uncataloged” volumes that are allocated after the last LOCAL_DATABASE backup, you must restore the catalogs and backup information. Otherwise skip to “Restore Data Created Before LOCAL_DATABASE Backup” on page 20-28.

Restore Catalogs and Backup Information

Restore all catalogs and backup information that were created or changed *and* backed up, staged, or part of a baseline backup after the LOCAL_DATABASE backup completed. (See Figure 20-2 on page 20-3.) This is restoring data from volumes that were allocated after the last LOCAL_DATABASE backup.

Note: This portion of the restore can also take a considerable amount of time.

1. Run a **dbreport volume**, sort the output and redirect to a file as follows:

```
emc# dbreport volume | sort > pre.sort
```

2. Use the EDM Library Unit Manager window in the EDM GUI to import all uncataloged volumes, when a backup was appended to a previous backup after the LOCAL_DATABASE was backed up.

Note: If you are doing a partial disaster recovery, and are skipping over portions of this chapter, be sure that the value of VM_ALLOW_DUP_SEQ_IMPORT in /usr/epoch/etc/vm/vm.cfg is set to "no," which is the default setting that allows the overwriting of a duplicate sequence number.

Look at which volumes are offline. Select all uncatalogued volumes and import them. When a message appears asking if you want to overwrite a duplicate sequence number, answer **yes**. This makes the latest backup to that trail available.

3. Run another **dbreport volume**, this time redirecting the sorted output to post.sort:

```
emc# dbreport volume | sort > post.sort
```

4. Run a **diff** on the two files:

```
emc# diff pre.sort post.sort
```

Examine the output for volumes that changed from "available" to "allocated." The following is an example of **diff** output that shows an EB volume that changed from "available" to "allocated:"

media	application	volume_name	seq	side	barcode	state	valid
DLT	EB	Server_alt_DLT	147	0	00000327	available	2BC53BEA44E275B4
DLT	EB	Server_alt_DLT	147	0	00000327	allocated	123456EA44E275B4

The second column indicates which EDM application uses the volume. EB indicates EDM Backup, EM indicates an HSM volume, and baseline indicates a baseline backup.

For EDM Backup volumes that changed, enter:

```
emc# ebimport -media valid -clevel 9 -level 9
```

For EM or baseline volumes that changed, enter:

```
emc# ebfs_import -v valid
```

Run **ebcatalogd -status** until it reports that all catalogs were processed.

At this point, you have restored all of your most recent server catalogs and backup information that were complete at the time of the last LOCAL_DATABASE backup and those completed after the last LOCAL_DATABASE backup.

Restore Data Created Before LOCAL_DATABASE Backup

You need to restore user files and some catalogs which were created before the LOCAL_DATABASE backup.

Before you begin this section, restart the ebcatalogd daemon if you did not do so in the previous section. To restart the daemon, enter the following command at the command line:

```
# /usr/epoch/EB/config/daemon_startup
```

1. Use **ebrestore** with the overwrite option set to never to restore each server work item you need to restore your filesystems and partitions, and any customized operating system files such as /.login, /.cshrc, /.profile, network databases, and crontab files as they were at the time of the last LOCAL_DATABASE backup.

```
# ebrestore -o never -i
```

The **-o never** option ensures that you do not overwrite any files that you already restored, such as / and /usr which were reinstalled in Step 2 on page 20-6. Overwriting them may crash the system.

CAUTION: You use the **-o never** option so that you do *not* restore all of / and /usr; since doing so may crash the system. For example, if you overwrite /usr/lib/libc.so.1 on a Solaris machine you crash the system.

However, you may want to restore some specific individual files selectively with custom settings (such as /etc/hosts and

**/etc/passwd) or programs loaded into
/usr/local or /usr/etc for local use using
ebrestore with overwrite set to always.**

At this point, you should have restored all filesystems and partitions that were present on the server up to the last LOCAL_DATABASE backup. (If you have an HSM system, EM and HSM now reattach properly.)

2. If an entire HSM VxFS filesystem was lost, you can do the following to restore their data quickly:

- a. Remount the filesystem without logging:

```
emc# mount -F vxfs -o remount,nolog /home1
```

- b. Perform the restore using **ebrestore** or the EDM Restore window.

- c. Remount the filesystem with logging when the restore is complete:

```
emc# mount -F vxfs -o remount,log /home1
```

3. If you restored any system files, reboot the server.

Note: Do not attempt multiple restores from the same trail simultaneously. This slows down the restores significantly.

Reenable crontab Entries

Reenable backups by undoing any edits you made to the root crontab file in “Disable Activity” on page 20-5 or “Stop Backups” on page 20-23.

If you do not want to restore past catalogs that allow you to restore older data, the recovery of the server is complete at this point.

Restore Past Catalogs

If you want to restore your earlier catalogs (which then allows you to restore earlier backup data), continue with this section. These backup catalogs provide a history of backups completed prior to the last LOCAL_DATABASE backup.

You know which catalogs were not yet restored because **ebreport history** may report savesets with ??????? in the Entries field. This indicates that the catalogs for those backups must be imported or restored before the backups can be restored.

1. Use **ebrestore** to restore the rest of the backup catalogs. Set the overwrite option (-o) to never.

```
emc# ebrestore -c emc -D emc -o never -d / -w emc:/data  
/data/epoch/EB/catalogs /
```

In this case, emc:data is the name of work item that backed up the partition on which the EDM Backup catalogs were stored, and /emc/data/epoch/EB/ is where the EDM Backup catalogs were stored (see the example in Figure 20-5 on page 20-9).

At this point you restored all of the backup catalogs up to the last LOCAL_DATABASE backup.

2. Run **ebexpire -partial -purge -expire** to eliminate any incomplete backups or catalogs that could confuse catalog processing.
3. Make sure all catalogs were processed.

- a. Run **ebcatproc -any** to start catalog processing for these particular savesets.
- b. Run **ebcatalogd -status**. Continue when **ebcatalogd** reports that all catalogs were processed.

ebcatalogd may indicate that some catalogs failed. If so, run **ebcatalogd -halt** to stop the catalog daemon and restart it with the **-retry_time 30** option, which causes these catalogs to be processed again in 30 seconds rather than the default value of one day.

If you make this change you should reset **ebcatalogd** after completion of the disaster recovery. To do this, enter:

```
emc# ebcatalogd -halt
emc# /usr/epoch/EB/config/daemon_startup -ebcatalogd
emc#
```

At this point, you restored the full EDM Backup catalog system (except for the most recent remote client catalogs which are discussed next).

4. Now, you can restore any backed-up data that was *not* present on the server at the time of the LOCAL_DATABASE backup, by using the **ebrestore -o never** command.

```
emc# ebrestore -o never
```

Restore Missing Catalogs

As a final step to restoring your data in a disaster situation you must make sure that you restored all of the missing backup catalogs. These backup catalogs were neither backed up by the work item for the partition containing the catalogs nor were they part of the LOCAL_DATABASE backup.

1. To determine which catalogs were not backed up prior to the LOCAL_DATABASE backup, run the **ebreport history** command with the **-ebimport** option.

```
emc# ebreport history -ebimport
```

This report displays ??????? in the Entries field if the saveset indicates the backup was complete but the catalog has not yet been restored.

In the example shown in Figure 20-10 on page 20-33, nine backups are missing their catalog files.

2. Identify the savesets that were not restored and import each one by using the following command:

```
emc# ebimport -clever 9 -level 9 -ok_if_unexpired saveset_id_1 ...  
saveset_id_n
```

This restores the catalogs even if the saveset database indicates they are not yet expired.

Figure 20-10

Output from ebreport history -ebimport

```

**** Work Items for Template Epoch_site, Primary Trailset ****
**Item "dilbert"

Time          Lvl ID          Status  Entries Expires  serverdb
1/ 3/98 20:56  0  5542FA53.2D28D220 sorted  ??????? 1/ 3/99
SaveSet ID 5542FA53.2D28D220

**Item "lamborghini"
Time          Lvl ID          Status  Entries Expires  serverdb
7/ 3/97 0:38  0  5542FA53.2C350E95 complete ??????? 7/ 3/99
5/31/97 19:20  0  5542FA53.2C0A93DB complete ??????? 5/31/99

**Item "odie"
Time          Lvl ID          Status  Entries Expires  serverdb
6/14/97 21:31  0  5542FA53.2C1D26F6 complete ??????? 6/14/90
5/17/97 19:40  0  5542FA53.2BF82217 complete ??????? 5/17/90

**Item "support"
Time          Lvl ID          Status  Entries Expires  serverdb
7/16/97 23:31  0  5542FA53.2C4773D6 complete ??????? 7/16/99
6/17/97 21:44  0  5542FA53.2C211F44 complete ??????? 6/17/99
5/20/97 19:42  0  5542FA53.2BFC1897 complete ??????? 5/20/99
4/30/97 19:49  0  5542FA53.2BE1BBA0 complete ??????? 4/30/99

Of the backups listed, 9 are missing catalog files.

Those backups whose count of catalog entries are filled with "?"
cannot be recovered until their catalogs have been recovered by running
"ebimport -ok_if_unexpired" on the backup's saveSet ID.

```

For information on restoring a UNIX client see Chapter 21
 "Recovering a UNIX Client from Disk Failure."

21 Recovering a UNIX Client from Disk Failure

This chapter contains disaster recovery procedures to use when you experience a disk crash on a UNIX client. Because each disaster is unique, these steps are presented only as a guide and not as all-inclusive, step-by-step instructions.

CAUTION: Performing a disaster recovery requires experience with EDM Backup administration (and HSM administration, if you have the HSM option), UNIX system administration, and the site environment.

In general, to restore a lost EDM client, you must replace the damaged disk, reinstall the operating system, reinstall (or relink) to any lost EDM client software, and use the EDM Restore window to restore your lost files.

If you also need to restore the backup server, be sure to restore the server first, *before* you restore the client. The following procedures assume that the EDM Backup server is not damaged or was already restored.

Note: This chapter applies to UNIX clients. Refer to the appropriate client supplements for recovering other clients.

Recovering a Client

Because you perform all of the restore in multiuser mode, you must notify the user community to stop all activity on the client. No one should be able to log in. The client must be inactive before you perform any disaster recovery procedures.

The following procedures also assume that all of the client filesystems are damaged or unavailable. If some filesystems remain intact, you might be able to skip some steps. These are guidelines only.

Beginning Steps

To restore an EDM client in case of a disk crash or major loss of files:

1. Edit root's crontab file on the server to comment out all backup and staging commands that might start up automatically. Comment out the following entries (note that some are HSM-specific (begin with em) and apply only to EDM Migration clients):

```
00 0 * * * /bin/kill -1 `cat /usr/epoch/etc/mal/emmasterd.pid` >/dev/null
2>&1#EPCmalc

15 23 * * * /usr/epoch/bin/emvck >/dev/null 2>&1#EPCmalc

00 1 * * * /usr/epoch/bin/emcompact -c >/dev/null 2>&1#EPCmalc

00 2 * * * /usr/epoch/bin/emscheck >/dev/null 2>&1#EPCmalc

00 3 * * * /usr/epoch/bin/emsundel >/dev/null 2>&1#EPCmalc

00 18 * * * /usr/epoch/EB/bin/ebbackup default >/dev/null 2>&1 #EPCebs

00 11 * * * /usr/epoch/EB/bin/ebexpire -expire -purge >/dev/null 2>&1 #EPCebs

00 1 * * * /usr/epoch/EB/bin/ebcatclean -fix_saveset >/dev/null 2>&1 #EPCebs

00 3 * * * /usr/epoch/EB/config/local_db_warning >/dev/null 2>&1 #EPCebs
```

2. Check hardware and replace if necessary. Run a disk diagnostics program. Be sure to identify and replace all damaged hardware *before* performing any software recovery procedures. Make sure replacement hardware is fully compatible with the system you had before the disaster. Each new disk should have at least the same storage capacity as the old disk.
3. Reinstall the client's native operating system. For directions, see the documentation that is supplied with the client. This reinstall process partitions your disk, restores the root filesystem on the client, and restores that portion of */usr* that the operating system loads. Make sure to specify that these filesystems are at least as large as and have the same (or larger) number of inodes as they had before the disaster.

CAUTION: Do not continue with software recovery until you are absolutely certain the disk is free of hardware problems.

If you are recovering a Backup client, go to step 6.

If you are recovering an HSM client, go to step 4.

For HSM Clients Only

4. If the client is an EDM Migration client, reinstall VERITAS VxFS filesystems. Customer Service has the *Software Installation* manual, call them with questions.

Go to step 6.

For Backup and HSM Clients

5. Reinstall the EDM Backup software using **eb_install_client**. Reconfigure EDM Backup on the client, using exactly the same settings as those defined prior to the disaster.

Refer to the full **ebreport disaster** report as necessary, for a list of the workitems prior to the disaster.

CAUTION: Do not restore all of / and /usr; doing so may crash the system. Only restore those files that contain customized settings; for example, /etc/hosts and /etc/passwd, or programs that are loaded into /usr/local or /usr/etc for local use.

6. Create and mount new, empty filesystems that are identical to (in terms of number of blocks and inodes), or larger than, those on the client before the disaster. Note that the commands used and command syntax vary from platform to platform. For a Sun workstation, for example, use **newfs**, **mkdir**, and **mount**. Refer to the **ebreport installation** or a full **ebreport disaster** report for a list of client filesystems and their sizes prior to the disaster.
7. If the filesystem that contained the backup software is still intact, but /usr was destroyed, you need to recreate the symbolic link from /usr pointing to this filesystem that contains the backup software.

If you are recovering a Backup client, go directly to step 12.

If you are recovering an HSM client, go to the next section, "For HSM Clients".

For HSM Clients

When recovering an HSM client, verify whether the EDM software (/ep_usr) is damaged.

- If it is not damaged, use the following steps in this order: step 9., on page 21-5 step 10., on page 21-5 and step 5., on page 21-4
- If it is damaged, use the following steps in this order: step 8., on page 21-5 step 10., on page 21-5 step 5., on page 21-4 step 11., on page 21-5 and then step 9., on page 21-5

8. Reinstall the EDM Migration (mc) software using **ep_install**.

9. Run **emlsconf** to display the old configuration. Any newly-created filesystems must be reconfigured for migration even if the MAL database files look intact. The commands are:

```
# emlsconf -r filesystem
# emfsconf filesystem old_parameters
```

where *old_parameters* are the **emfsconf** parameters displayed by the **emlsconf** command.

10. Shut down the HSM daemons prior to restoring the EDM software filesystem:

```
emc# sh .usr/epoch/etc/rcK/K60mal stop
```

11. Restore the filesystem where the EDM software was installed (/usr/epoch/etc) out of place, and then move it to the proper location (/usr/epoch) to bring back the database files. Then reboot the migration client.

12. If you have an EDM database client, reinstall the vendor database software, if needed, via their instructions. Then reinstall the Backup software, if needed. For offline database backup and Oracle online backup, you can do this through the EDM Backup Configuration window. For earlier online backup clients, see the appropriate online backup supplement and release notes.

For Both Backup and HSM Clients

After you reinstall the EDM Backup software using **eb_install_client**, you are ready actually to restore your files.

13. Open the EDM Restore window:

- from the client using:

```
client1# edmcrestore
```

or

- from the EDM Main window.

Use the EDM Restore window or **ebcrecover** to restore your lost filesystems and files. In the EDM Restore window you can browse a list of work items for the client, select the work item, mark the files you want to restore, and start the restore.

14. Reboot the client system.

Part VI

Appendixes

A **Directory Structure**

This appendix describes the directory structure of EMC Data Manager backup, volume management, and HSM.

This is the directory structure of the software that runs on a server and the software that runs on the client, which enables the client to collect backup data and restore recovered files.

This appendix describes the following:

- Backup Server Directory Structure
- Backup Client Directory Structure
- Volume Management Directory Structure
- HSM Directory Structure

Backup Server Directory Structure

The EDM backup server software handles all aspects of backup and restore operations. Its components are:

- EDM transfer protocol or remote shell (**rsh**) – Communication between the server and client. The EDM transfer protocol is supported on UNIX platforms.
- EDM Backup server processes – Daemons and programs that service backup and restore operations.
- The Global Configuration Database – Configuration specifications that direct backup and restore operations and files that manage the database.
- The Installation Directory – Directory on the EDM server that contains backup installation and executable files for both the server and the client systems. This section describes backup by looking at the elements on the server that operate the backup and restore operations.

EDM stores two sets of files on the server: large and small. The set of large files (backup catalogs and log files) grows over time and the set of small files (client binaries and configuration files) does not.

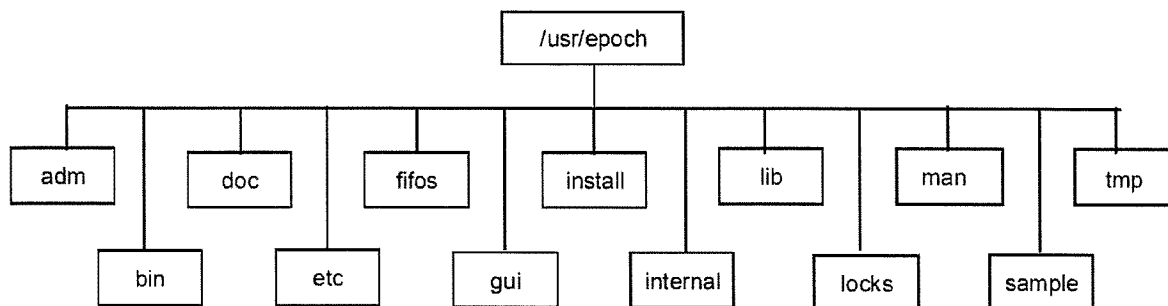
/usr/epoch

The /usr/epoch directory includes several subdirectories that contain configuration and database files.

CAUTION: Although these files are text files, you should never attempt to modify them with an ordinary editor. The configuration commands and the EDM Backup Configuration window do more than just modify the files.

Figure A-1 shows many of the top-level directories.

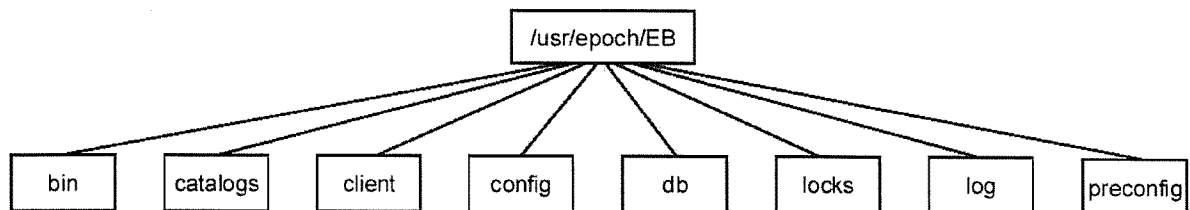
Figure A-1

Server Directory Structure (/usr/epoch)

- adm – location for circular and archived log files
- bin – contains user executable binaries
- doc – contains online documentation
- etc – contains vital EDM software databases
- fifos – EDM software fifos
- gui – gui resource files
- install – contains non user executable install files
- internal – contains EDM internal commands
- lib – contains shared libraries and non user executable daemons (linked to from /usr/epoch/etc/lm)
- locks – contains lock files
- man – contains all man pages
- sample – contains sample Volume Manager and Library Manager configuration files
- tmp – EDM software tmp directory

/usr/epoch/EB

The next level of backup server directories, /usr/epoch/EB, contains the following subdirectories.

Figure A-2**Server Directory Structure (/usr/epoch/EB)**

- bin – contains the server executables
- catalogs – contains the backup catalogs and the saveset database (ebsaveset_db)
- client – contains client executables and installation files
- config – contains the configuration files, including eb.cfg
- db – contains database files; for example, cattask.list and htab
- locks – contains lock files; for example, ebcatalogd.lock
- log – contains the backup log files
- preconfig – contains a file for each client with default configuration information

When you initially configure backup using **eb_server_config**, you can create the directory /usr/epoch/GENDIR/EB to hold large catalogs and log files.

If you run **eb_server_config** without the **-D** option, you are prompted to create the alternate directories. **eb_server_config** also creates the following symbolic links (symlinks):

- /usr/epoch/EB/catalogs is a symlink to /usr/epoch/GENDIR/EB/catalogs
- /usr/epoch/EB/log is a symlink to /usr/epoch/GENDIR/EB/log

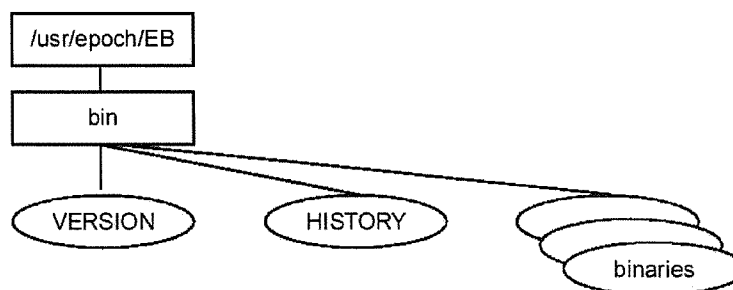
Refer to the **eb_server_config** man page for more information.

Bin Directory

The bin directory contains all the server binaries and a VERSION file that contains the current version of EDM Backup software and a HISTORY file that contains a history of all EDM Backup installations.

Figure A-3

The bin Directory



Catalogs Directory

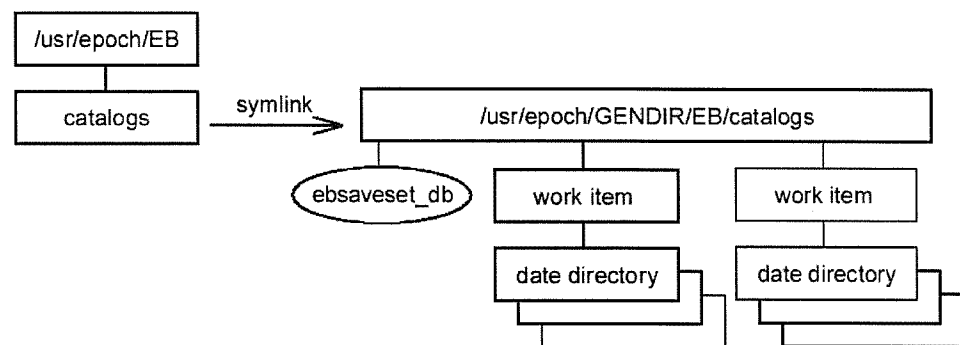
The catalogs directory contains the saveset database and one directory for each configured work item. This subdirectory has the same name as the work items and is created as necessary by the catalog subsystem. Any "/" characters in the work item's name are translated to "%" characters for use in directory and filenames.

Below the work item directory are DATE directories. DATE directories are named with four-year digits followed by a dash and the month digits (for example, 1998-02). The catalog software must be able to create a date directory when it writes a new backup catalog.

The purpose of this directory structure is to enable system administrators and customer support personnel to scan directories of moderate size for catalog files, while allowing the software a simple and fast algorithm for locating catalogs.

Figure A-4

The catalogs Directory



Client Directory

The client directory contains binaries and other client platform-specific software. Under this directory are client platform directories of the form:

manufacturer_cpu_os[_other]

The *_other* suffix is optional. It is used for anything that does not fit into the scheme of manufacturer, cpu, and OS. Normally, it is not used. Client platform directories have names such as *sun_sun4_v5* or *hp_800_v10*.

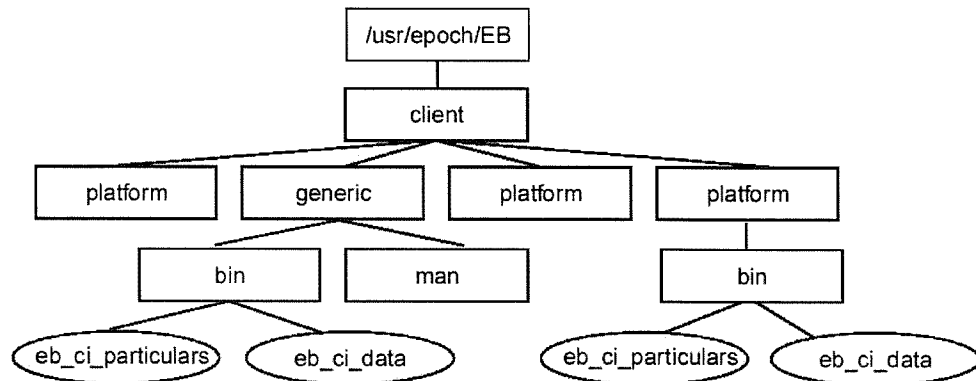
The client directory also contains a subdirectory called generic. This subdirectory (as well as each of the platform subdirectories) contains a bin subdirectory, which contains up to two files that are used in customizing the client installation process. The file that is shipped as part of the backup software distribution package is named eb_ci_particulars. The second file is generated from the first and is named eb_ci_data. The following describes each file:

- eb_ci_particulars file – Contains definitions set by EDM for constants that should not have to be changed at your site (such as the name of the platform-specific options to be used with the **mmtpts** command).
- eb_ci_data file – Contains the generic platform's constants. This file is created during backup installation for each platform type under /usr/epoch/EB/client.

Under each specific client platform directory are the subdirectory, bin and optionally, the subdirectory man. The bin directory contains all client scripts and binaries for installation of the client as well as those for the normal backup operation of the client. Like the bin subdirectory under generic, these bin directories may also contain the eb_ci_particulars file. In all cases this subdirectory contains the eb_ci_data file. The man directory contains man pages for the client recovery program(s).

Figure A-5

The client Directory

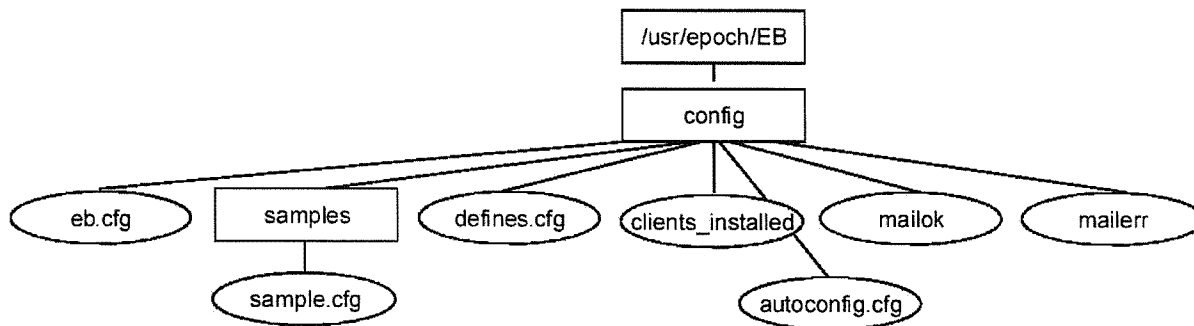


Config Directory

The config directory contains the default `eb.cfg` configuration file, sample configuration files, and a sample **findxcpio** macro file, called `defines.cfg`. It also contains a file called `clients_installed` which is modified when each client is installed. It includes the **mailok** script to which the backup software passes backup completion reports, and the **mailerr** script to which it passes backup failure reports.

Figure A-6

The config Directory



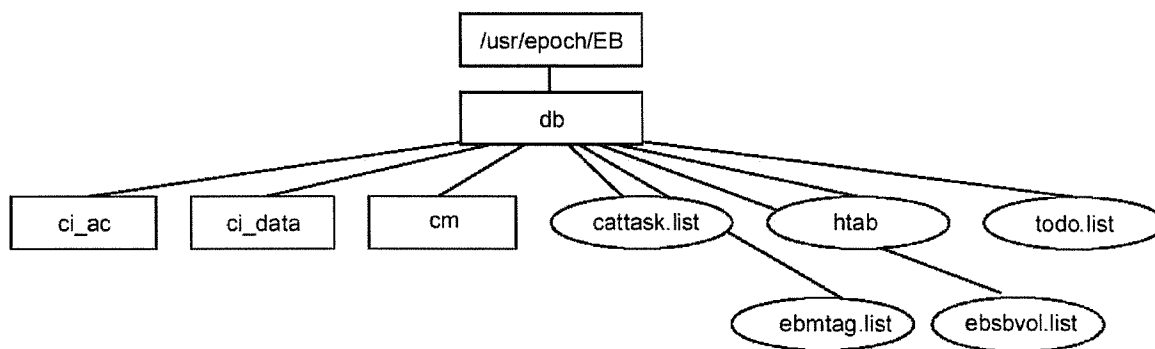
If you installed backup to backup all (or some) clients by using the autoconfig facility, this directory also contains a default configuration file for each such client (`/config/client-name/dcfg`), and an `autoconfig.cfg` file. This `autoconfig.cfg` file references the default (`dcfg`) file for each client, via an `#INCLUDE`, and provides a single work-group definition for the `auto_configured_work_group`. You can then create a template that references that work item and thus backs up all the autoconfigured clients at once.

Db Directory

The db directory contains the EDM backup database files.

Figure A-7

The db Directory



- `ci_ac` is a subdirectory that contains information that is used by `recover`'s graphical user interface. It stores information similar to that in the `preconfig` directory but is more tuned toward graphical standards.
- `ci_data` is a subdirectory that stores the installation parameters for each client machine. These parameters describe what occurred during the most recent client installation. One file in this subdirectory exists for each machine installed, and it is named after the machine:
`/usr/epoch/EB/config/db/ci_data/client-name`.

- `cm` is a subdirectory that contains all of the information that the **coverage** report uses.
- `cattask.list` is **ebcatalogd**'s database file. It contains all of the information that is needed to process the catalogs that **ebbackup** generates.
- `htab` lists and identifies all of the installed clients.
- `todo.list` is **ebbackup**'s database file. It contains all of the backup history, including when the backups occurred, how long they took, the old and current schedules, etc.

The following files exist only on an HSM system:

- `ebsbvol.list` (the saveset-to-baseline relations file) is used for baseline backups (levels B1 and B2). For each physical volume that is used in a baseline backup, this database records the volume ID and saveset ID of the backup.
- `ebmtag.list` maps work item names to EpochMigration tags.

Locks Directory

The locks directory contains lock files, for example, `ebcatalogd.lock` and a lock file for each work item. This directory is flat; no subdirectories exist under locks.

Log Directory

The log directory (which may contain a symbolic link to `/usr/epoch/GENDIR/EB/logs`) contains all of the EDM backup logs files. For more information on log files, see page 16-35.

Preconfig Directory

The preconfig directory contains a file for each client. This file contains default configuration information for the client.

Server Man Directory

The server man pages are installed in `/usr/epoch/man`.

For a listing of both server and client man pages, see Chapter 18 "Man Page Listing".

Table of Backup Server Directories and Files

Table A-1 describes each of the server's configuration files.

Table A-1

The Server's Configuration Directory

File or Directory Name	Description
/usr/epoch/EB	Contains the server's EB directories and files.
/usr/epoch/EB/bin	Contains the server's binaries.
/usr/epoch/EB/bin/VERSION	Contains the current installed EDM backup version.
/usr/epoch/EB/bin/HISTORY	Contains the EDM backup installation history.
/usr/epoch/EB/catalogs	May contain a symbolic link to the backup catalog directory on the EDM backup server: /usr/epoch/GENDIR/EB/catalogs This stores the backup catalogs.
/usr/epoch/EB/client	Contains all client installation files. Contains a subdirectory for each platform manufacturer, CPU architecture, and OS revision supported for EDM backup clients.
/usr/epoch/EB /client/ <i>manuf_cpu_os</i>	Describes the platform manufacturer, CPU architecture, and OS revision. Each subdirectory name is of the format: manufacturer_cpu_os (e.g., sun_sun4_v5). This directory is not used until the client installation procedure is performed.
/usr/epoch/EB /client/ <i>manuf_cpu_os</i> /bin	Contains platform manufacturer client binaries.
/usr/epoch/EB /client/ <i>manuf_cpu_os</i> /man	Contains the client man pages.
/usr/epoch/EB/client/generic/bin	Contains binaries that are used to customize client installations.
/usr/epoch/EB/config	Contains the EDM backup configuration files.
/usr/epoch/EB /config/autoconfig.cfg	Contains information used by autoconfig to define the auto_configured_work_group, and to reference the default configuration file for each autoconfig client.
/usr/epoch/EB /config/ <i>client-name</i> /dcfg	Describes the configuration for each client that is backed up via the autoconfig facility.

Table A-1

The Server's Configuration Directory (Continued)

File or Directory Name	Description
/usr/epoch/EB/ config/clients_installed	Contains an entry for each EDM backup client installed. You can read but not edit this file.
/usr/epoch/EB/config/defines.cfg	Contains a sample findxcpio macro file.
/usr/epoch/EB/config/mailerr	Contains the script to which the backup software passes backup failure reports.
/usr/epoch/EB/config/mailok	Contains the script to which the backup software passes backup completion reports.
/usr/epoch/EB/config/samples	Contains sample configuration files to use as a reference when editing the eb.cfg file.
/usr/epoch/EB/config/eb.cfg	Contains the server configuration file. You edit this file by way of the EDM Backup Configuration window to specify the server's configuration information.
/usr/epoch/EB /config/local_db_startup	Dictates what gets backed up by the LOCAL_DATABASE backup.
/usr/epoch/EB /config/local_db_cleanup	Dictates what happens after the LOCAL_DATABASE backup is done.
/usr/epoch/EB /config/local_db_warning	Mails a warning message to the administrators if the LOCAL_DATABASE backup is too old.
/usr/epoch/EB/db	Contains ebbackup 's database files.
/usr/epoch/EB/db/ci_ac	Contains information used by the EDM Restore window.
/usr/epoch/EB/db/ci_data	Stores the installation parameters describing what took place during the most recent installation of each client machine.
/usr/epoch/EB/db/cm	Contains all the information used by the coverage report.
/usr/epoch/EB/db/ebmtag.list	Maps work item names to EDM Migration tags.
/usr/epoch/EB/db/todo.list	Contains all the backup history (when backups occurred, how long they took, schedules, etc.).
/usr/epoch/EB/db/ebsbvol.list	Stores the volume id(s) and saveset ID for each baseline backup.

Table A-1

The Server's Configuration Directory (Continued)

File or Directory Name	Description
/usr/epoch/EB/db/cattask.list	Contains all information needed to process the catalogs generated by the backup software.
/usr/epoch/EB/locks	Contains all of the temporary lock files that the backup software creates during backup and restore operations. This includes locks on templates and trailsets.
/usr/epoch/EB/log	May contain a symlink to the log file directory on the EDM backup server: /usr/epoch/GENDIR/EB/logs Stores the server log files.
/usr/epoch/EB/preconfig	Contains a file for each client; each file contains default configuration information for the client.
/usr/epoch/man	Contains the man and cat directories for the man pages.

Backup Client Directory Structure

All backup files on the backup client are stored in the backup client's home directory.

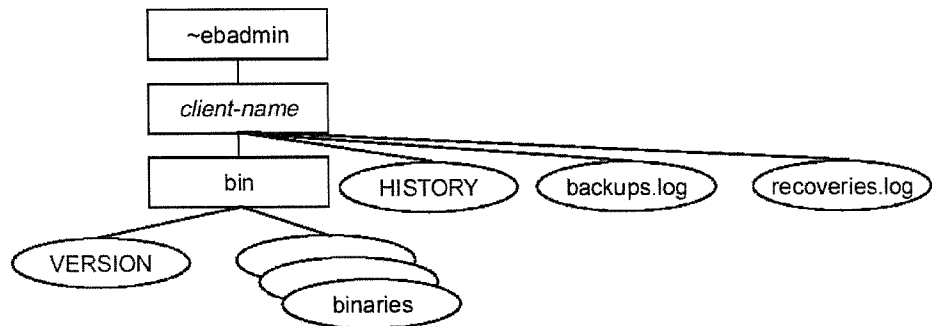
The examples in this document specify the name of the default user, ebadmin. The home directory is specified as ~ebadmin.

Note: For security purposes, ebadmin should not have a password, and a user should not be able to log in. The backup software creates the default user name with no password. (Refer to "Security Issues" below.)

Client Home Directory

The goal of the client installation procedure is to set up the backup client home directory as shown below. Assuming that you used the default backup client username ebadmin, the home directory is ~ebadmin. You can have a separate backup client home directory on each client system or you can specify

one directory that all the clients mount via NFS. The directory structure allows this type of sharing, although it does not allow one client to be backed up by multiple EDM units.

Figure A-8**The ~ebadmin Directory**

Under the home directory there is a subdirectory for each client, named with the client's network host name. The client directory contains the bin directory which holds all the client binaries as well as a VERSION directory that displays the versions of the binaries. The client directory also contains the backups and recoveries log files (for more information, see "Log Files" on page 16-35) and a HISTORY file that displays each client installation.

In the case where many clients share the same platform and OS release, installing binaries in each client's home directory is redundant. Thus, you can replace a bin directory with a symbolic link to a bin directory of another client of the same type. You must do this manually (it is not done during installation).

If you reinstall or deinstall a client, the existing bin directory is removed. Thus, any clients that have links to that directory are not backed up.

Security Issues

A potential security hole exists for backup client directories that are shared via NFS by more than one client. For example, consider two clients, client1 and client2, that share the same backup client home directory. Call the backup client user ebadmin and this home directory ~ebadmin. In this case, someone who is logged in as root on client1 can go into the ~ebadmin directory on client1 and change its .rhosts file to allow access to the directory as ebadmin from client1 (normally access is only allowed from the EDM). He or she can then contact client2 as ebadmin from client1 and cause client2 to perform backups of any file on client2 and send the output to client1. Furthermore, the backup data can then be manipulated on client1 and sent back to client2 as a restore so as to overwrite existing files on client2.

Note: If this type of security is of concern to you, do not share client backup home directories via NFS.

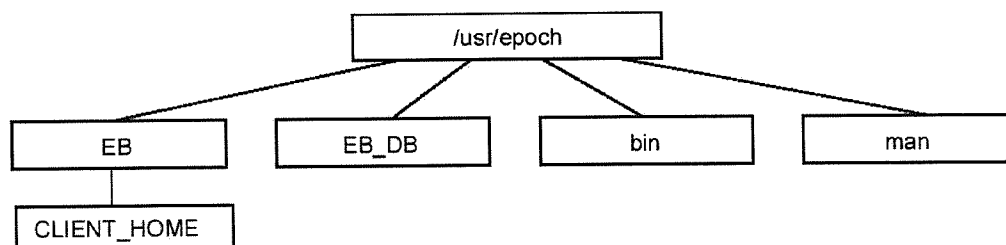
Additional Client Software

The default location for ~ebadmin is /usr/epoch/EB/CLIENT_HOME.

Some of the other client directories are listed in Figure A-9.

Figure A-9

Client Directory Structure



- EB – contains the client home directory (~ebadmin)
- EB_DB – holds database backups
- bin – contains report and restore scripts
- man – contains client man pages

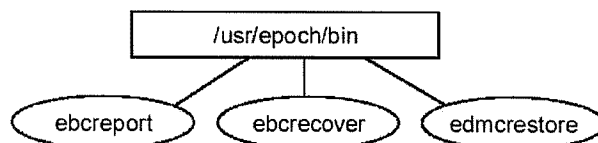
Client EB_DB Directory

Each installed client gets the /usr/epoch/EB_DB directory which holds database backups.

Client bin Directory

You can also install scripts on selected clients for client file self-recovery and report generation. By default, the scripts are installed in /usr/epoch/bin. You can also install the scripts in an alternate directory.

Figure A-10

The Client bin Directory

Client man Directory

You can choose to install the client man pages in either /usr/epoch/man, /usr/local/man, or in a location of your choice.

For a listing of both server and client man pages, see Chapter 18 "Man Page Listing".

Table of Backup Client Directories and Files

Table A-2 describes these client files and directories.

Table A-2

Client Directories

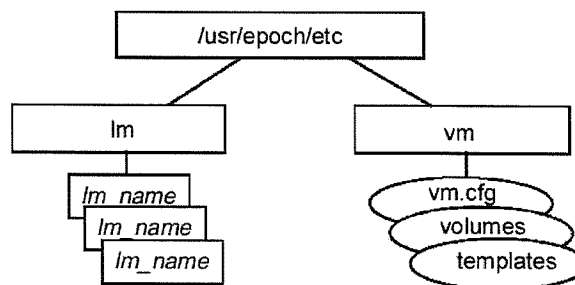
File or Directory Name	Description
~ebadmin (/usr/epoch/EB/CLIENT_HOME)	Default client home directory that contains a subdirectory for each client, named with the client network host name.
~ebadmin/ <i>client-name</i> /bin	All client binaries for the client are stored here together with a VERSION directory showing the version of these binaries. The bin subdirectory can be a symbolic link to the bin subdirectory of another client with the same architecture and OS.
~ebadmin/ <i>client-name</i> /HISTORY	HISTORY file shows each installation that is performed for the client.
~ebadmin/ <i>client-name</i> /backups.log	Client's backups.log file records critical information about backups performed on this client.
~ebadmin/ <i>client-name</i> /backups.log.cminfo	Contains coverage monitor data that is sent to the server after each backup.
~ebadmin/ <i>client-name</i> /recoveries.log	Client's recoveries.log file records information about file restores that are performed on this client.
~ebadmin/ <i>client-name</i> /eb_ci_data	Contains shell script values that are required by various shell scripts such as startfind .
/usr/epoch/man or /usr/local/man (or an alternate directory)	Contains the client's man pages.
/usr/epoch/bin/ebcreport	Enables networked clients to produce reports such as media and backup history reports. (See man page.)
/usr/epoch/bin/ebcrecover	Allows a networked client to restore files, directories, and filesystems. (See man page.)
/usr/epoch/bin/edmcrestore	Starts the EMC Data Manager Restore window from a networked client.
/usr/epoch/EB_DB	Contains files for backups of database files.

Volume Management Directory Structure

The `/usr/epoch/etc` directory has many subdirectories that contain vital EDM software database files. Two of these are the Library Manager and Volume Manager files. Figure A-11 shows these two subdirectories of `/usr/epoch/etc`.

Figure A-11

`/usr/epoch/etc` Directory



Library Manager Subdirectories

Each Library Manager resides in an individual subdirectory in `/usr/epoch/etc/lm`. For more information, see “Library Manager Configuration Files” on page C-9. The directory name matches the Library Manager name that appears in the Display area of the main volume management window.

Within each Library Manager directory, **lmconfig** creates several files. During start-up, the Library Manager reads the files in its directory to initialize the library unit it is managing and to set up its internal data structures. Table A-3 describes the files that are located in each of the Library Manager directories.

Table A-3**Files in /usr/epoch/etc/lm/lm_name**

File	Description
lm.cfg	Configuration file for the Library Manager. Library Manager configuration includes parameters that define the hardware address (SCSI bus, target ID, and lun) of the device, Library Manager name, number of drives, and scheduling parameters for the robot and drive(s). See "Library Manager Configuration Files" on page C-9.
volid.dat	Inventory list of the library unit's contents. This file enables the Library Manager to start up without taking a complete inventory of the library unit. If volid.dat does not exist, the Library Manager inventories the library unit and creates the file. The Library Manager updates the volid.dat file when it: <ul style="list-style-type: none"> • writes a label to a volume • moves a volume into or out of an inlet, slot, or drive • enables or disables a library unit slot • takes an inventory of the library unit
dn	Soft link to the physical name of the drive that is located in /devices. <i>n</i> indicates the drive number.
lm_in_drive_n.dat	Drive contents file for each drive installed in the library unit; where <i>n</i> indicates the drive number.
clog	Circular log file that contains a detailed description of the activity for this Library Manager.
lmd	Link to the Library Manager executable daemon located in /usr/epoch/lib/rvm.
liblm_tnumber	Contains a number used internally to identify the Library Manager.
lm_is_open	Lock file that ensures that only one copy of the Library Manager daemon is running at one time.

Volume Manager Files

The directory `/usr/epoch/etc/vm` contains all files that are related to the Volume Manager. For more information, see “Volume Manager Configuration File” on page C-2. Table A-4 describes the files that are contained in `/usr/epoch/etc/vm`.

Table A-4**Files in `/usr/epoch/etc/vm`**

File Name	Description
clog	Volume Manager daemon's circular log (clog) file. An ascii file that serves as a debug log and contains a detailed description of the vmdaemon's activity.
notd	Link to the notify daemon located in <code>/usr/epoch/lib/rvm</code> .
notd.pid	Notify daemon's process ID file.
notd_clog	Notify daemon's circular log (clog) file. An ascii file that serves as a debug log for the notify daemon. It contains a detailed description of the notd's activity.
templates	Template catalog contains volume templates for labeling media.
templates.def	Template record definition file.
templates.templateid.ndx	Template index file.
vm.cfg	Volume Manager configuration file. This file contains parameters that define the location of the vmdaemon, set the message-logging level, and lists the Library Managers configured for this server. See “Volume Manager Configuration File” on page C-2.
vm_is_open	Lock file which prevents two Volume Manager daemons from running at one time.
vmdaemon	Link to the Volume Manager executable daemon that is located in <code>/usr/epoch/bin</code> .
vmerd	Link to the Volume Manager's erasing daemon that is located in <code>/usr/epoch/lib/rvm</code> .
volumes	Volume catalog. contains complete volume information for the server.
volumes.def	Template catalog contains volume templates for labeling media.
volumes.volid.ndx	Template index file; an internal file used exclusively by the vmdaemon.

HSM Directory Structure

The `/usr/epoch/etc/mal` directory includes several subdirectories which contain server and client configuration and database files. This section provides detailed information on this directory structure.

HSM Configuration Database

Every EDM HSM server and every migration client contains a migration configuration database. This database consists of structured text files that are updated by the **emstconf**, **emfsconf**, and **emsysconf** commands and by the functions you perform when using the EDM HSM Configuration window.

The database contains information that specifies which filesystems are stageable, when files should be staged, and the staging templates to which filesystems are assigned.

CAUTION: Although these files are text files, you should never attempt to modify them with an ordinary editor. The configuration commands and the EDM Backup Configuration window do more than just modify the files; they also know how to interact correctly with any staging processes that are running.

On client systems, the database also lists the fileserver and store to which staging templates are assigned.

The text files are stored in the `/usr/epoch/etc/mal/` directory.

Figure A-12

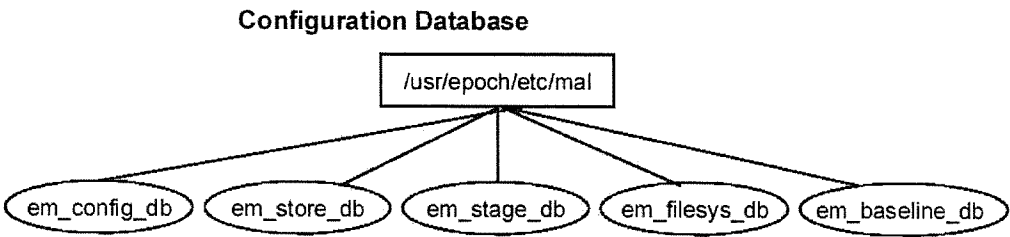


Table A-5 lists the database files.

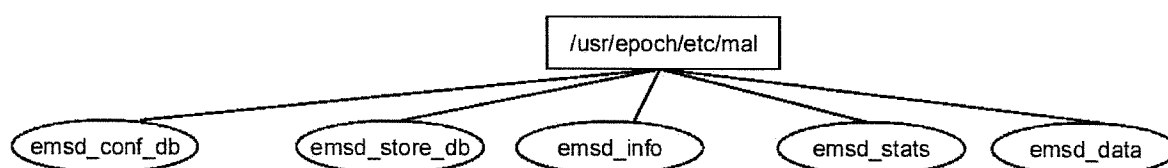
Table A-5 Migration Database Files

Argument	Description
em_config_db	Text file that contains system-wide configuration data. You update this file with the emsysconf command or when you change global properties with the Configuration Interface.
em_stage_db	Text file that contains staging template configuration data. You update this file with the emstconf command or when you change staging template information with the Configuration Interface.
em_store_db	Text file that contains client store information. You update this file with the emstconf command or when you change store information with the Configuration Interface.
em_filesys_db	Text file that contains per-filesystem configuration data. You update this file when you issue the emfsconf command or when you change filesystem information with the Configuration Interface.
em_baseline_db	Text file that contains baseline backup information. This data is manipulated by the HSM software on behalf of the backup software.

Network HSM Server Database The network migration server has a global configuration database that contains information about network migration activity and the default client store values. The global configuration database files reside in /usr/epoch/etc/mal.

CAUTION: Editing these files directly may result in loss of data.

Although both the emsd_conf_db and the emsd_store_db files contain editable text descriptions of the configuration, do not edit these files directly. Instead use the server's configuration commands or the EDM Backup Configuration window to make any modifications to the database. There are five database files.

Figure A-13**Global Database Files****Table A-6****Global Database Files**

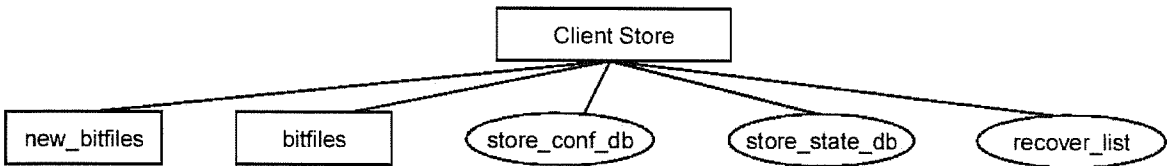
File	Description
emsd_conf_db	Text file that defines the limits on the HSM software protocol requests and the default values for client store configurations.
emsd_info	Binary file that contains information about the currently executing emsd process.
emsd_store_db	Text file with a list of configured client stores and their locations on the server.
emsd_stats	Binary file that contains cumulative statistics on HSM software protocol traffic and client agent activity.
emsd_data	Binary file that contains the HSM usage history on the server.

Client Stores

Each client store has its own file hierarchy and is logically independent from every other client store. The client store's top-level directory contains three files and two subdirectories.

As system administrator you see these files and directories when you list the contents of the client store directory.

Figure A-14 **Client Store Organization**



The client store's top-level files and directories are listed in Table A-7.

Table A-7 **Client Store Files and Directories**

File/Directory	Description
store_conf_db	Text file of the store's configuration information.
store_state_db	Text file of the store's state information that the client agent keeps current.
recover_list	List of the bitfiles to restore from the EDM server's backups.
new_bitfiles	Temporary holding directory for bitfiles that are being created as part of a stage out from a client system.
bitfiles	Directory that contains the completed bitfiles in a 3-level hierarchy.

When the client agent creates a bitfile, it gives the bitfile a 16-digit hexadecimal name and places it in the new_bitfiles directory. The bitfile remains there until it is completely written. Once the bitfile is complete, the client agent moves it to the bitfiles directory.

B EDM Backup Configuration File

The backup configuration file (`/usr/epoch/EB/config/eb.cfg`) on the EDM Backup server defines how backups run at your site. The configuration interface stores its data in the `eb.cfg` file.

You configure your EDM backups using the Backup Configuration wizard, and you tailor your configurations to your needs by using the EDM Backup Configuration window. (Avoid manually editing the configuration file.) When you do not have access to the graphical interface (such as when dialing in), you can use the interactive command line interface, `eb_config`.

WARNING: Manually editing the `eb.cfg` file risks corrupting the backup configuration.

This chapter describes the statements within the configuration file. These statements correspond to windows and fields in the EDM Backup Configuration wizard and window, but they do not match exactly in wording or in number. This chapter does not describe Symmetrix Path, Symmetrix Connect, or new database clients.

The topics in this chapter include:

- General Coding Rules
- Summary of Fields
- Server Fields
- Work Group Fields
- Filesystem Work Item Fields
- Database Work Item Fields
- PC Work Item Fields
- Backup Trailsets

General Coding Rules

If you must edit the configuration file, follow these general editing rules:

Note: The EDM Backup Configuration window reformats the eb.cfg file. If you edit eb.cfg directly, you lose comments and spacing the next time you run the interface. The EDM Backup Configuration window reads in C pre-processor #include statements.

- The configuration file is made up of nested sets of blocks; a set of curly braces ({ ... }) delimits each block. Make sure to use these braces in pairs. For each open brace, there must be a closing brace.
- If you include multiple conflicting specifications for any given field, the last specification is used in most situations, unless otherwise noted in this chapter.
- Use a semicolon (;) to terminate entries that do not begin with a brace:
`work group list: "dave's group","tony's group";`
- Separate multiple strings by using a comma and optional spaces:
`authorized backup list: "cad1", "cad2", "cad3",
"atlas2";`
- Do not use the line-continuation character (\).

- Any comments are delimited using paired slash-asterisk characters: */*comments*/*. Comments should not span lines.
- Place double quotes around *strings*, where a string is any non-numeric value except the reserved words that are described in the next bullet.

```
client backup username: "ebadmin";
```

Each quoted string must be complete on a single line: strings cannot span lines. Make sure to specify each string by using the correct case (capitalization). EDM Backup is case-sensitive when interpreting strings.

- Some reserved words do not require quotes; you can enter them as follows:
 - Always use the standard three-character abbreviation when specifying a month. Use all lowercase characters, or uppercase the first character only (for example, *mar* or *Mar*).
 - Specify units of time as *hours*, *minutes*, *seconds*, *days*, *weeks*, etc., as described for each field. Always use lowercase characters when specifying units of time.
 - Spell out days of the week completely. Use all lowercase characters, or uppercase the first character only (for example, *sunday* or *Sunday*).
 - Specify units of computer storage as follows: KB or K; MB or M; or GB or G (lowercase or uppercase).

Checking Your Changes

If you do edit the configuration file, you always must check afterward for syntax errors. To do this, run **ebbackup** with the **-check** option. This causes **ebbackup** to report any syntax errors.¹

1. Under certain unusual conditions, some syntax errors will not be detected until the actual backup is run.

As necessary, edit the file to correct the errors, and then run the check again. Repeat this process until no more errors are present.

When Changes Take Effect

When you change various configuration parameters, changes either:

- take effect immediately on your system (even as backups are being processed), or
- take effect the next time **ebbackup** processing is started, either from an entry in root's crontab file or by issuing the **ebbackup** command manually

This does not include the next time **ebbackup** autoscheduling-only (**-sched** option) or checking (**-check** option) is started. Any exceptions are noted in the discussions.

Summary of Fields

Table B-1 summarizes the fields in the configuration file. It includes a brief description of each field, then tells whether the field is required (Req column), the field length and range of possible values, the initial value at installation, and when a change to the field takes effect.

The fields are presented generally in the order that they appear in the file.

Table B-1

Summary of Fields in the Configuration File

Field	Description	Req	Size or Values	Initial Setting	Takes Effect	
					Now	Next Bkup
Server Block Fields						
ebserver	Name of the EDM Backup server	yes	1-63 char	server-name	—	—
client backup username	Login name used when ebbackup connects with client system	yes	1-63 char	ebadmin		✓
backup administrator usernames	Individuals who are responsible for EDM Backup, and who have root-like privileges within the EDM Backup environment	yes	1-63 char (per user)	root	✓	
authorized backup list	Client systems that are backed up by the EDM Backup server	yes	1-63 char (per client)	none	✓	
authorized recovery list	Users who can restore files backed up from their own client (known as a <i>self-service restore</i>)	no	<i>client:user</i> combinations	none	next restore	
authorized cross recovery list	Users who can restore files to clients other than those from which the backup files originated	no	<i>client:user</i> combinations	none	next restore	
recovery administrator list	Users who can restore files other than their own to the client from which the files were originally backed up	no	<i>client:user</i> combinations	none	next restore	
maximum simultaneous clients	Global maximum number of clients that are backed up concurrently across all trails	yes	in the range 1-64	24	✓	

Table B-1

Summary of Fields in the Configuration File (Continued)

Field	Description	Req	Size or Values	Initial Setting	Takes Effect	
					Now	Next Bkup
use at most... trails concurrently	Maximum number of trails EDM Backup can write to a specific type of media concurrently	yes	<i>nn media_type</i>	4 per type of device	✓	
limit throughput	Total throughput of backup processing	yes	no limit <i>nnn K</i> per hour <i>nnn K</i> per minute <i>nnn K</i> per second (<i>K</i> is KB, MB, or GB)	600KB per second	✓	
maximum server backups.log file size	Maximum size of the server's backups.log file	yes	<i>nnn K or</i> no limit (<i>K</i> is KB, MB, or GB)	256KB	✓	
maximum server recoveries.log file size	Maximum size of the server's recoveries.log file	yes	<i>nnn K or</i> no limit (<i>K</i> is KB, MB, or GB)	256KB		next restore
maximum client backups.log file size	Maximum size of each client's backups.log file	yes	<i>nnn K or</i> no limit (<i>K</i> is KB, MB, or GB)	64KB		next backup for the client
maximum client recoveries.log file size	Maximum size of each client's recoveries.log file	yes	<i>nnn K or</i> no limit (<i>K</i> is KB, MB, or GB)	64KB		next backup for the client
catalog threshold to force level 0 backup	Level 0 backup is forced for any filesystem work item when more than this number of files in the work item have never been backed up	no	number of files; 0 disables this feature	0		✓
Work Group Field						
work group	Name of the work group	yes	1-63 char	none		✓
File System Work Item Fields						

Table B-1

Summary of Fields in the Configuration File (Continued)

Field	Description	Req	Size or Values	Initial Setting	Takes Effect	
					Now	Next Bkup
work item	Name of the work item, the client backed up by the work item	yes	1-63 char (each, per work item & client name)	none		✓
filespec	Name of one or more client directories, filesystems, or files that receives level 0-9 backups via this work item	yes	pathnames and optional findxcpio qualifiers (no size limit)	none		✓
baseline filespec	Same as above, but lists those local-client directories and filesystems in HSM systems that receive a baseline backup	no	same as <i>filespec</i>	none		✓
migration backup tag	Tag used to cross-reference a migration store on the server with the migration-client files from which the store was created (applicable when backing up staged files from the server)	no	1-63 char; first 16 must be unique	none		✓
exclusion tag	Marker that identifies work items that should not be backed up concurrently	no	1-63 char	none		✓
connection via	Specifies the use of an alternate network port for a work item in a multiple networked client.	no	alternate port name, must be a valid client name	none		✓
priority	Priority at which the work item should run	no	in the range -25 (run first) to 50	0		✓
do not load balance	Statement used to exclude the work item from load balancing (so no extra level-0 backups are taken)	no	as stated	statement omitted (load balance)		✓

Table B-1

Summary of Fields in the Configuration File (Continued)

Field	Description	Req	Size or Values	Initial Setting	Takes Effect	
					Now	Next Bkup
completeness	Code used to limit the number of files for which the data portion is written to the backup media (applicable for files under migration control, to avoid performing excessive backups)	no	all files resident files only files not backed up in migration store non-baselined files only	varies by client		✓
level map	Mapping between backup levels that normally occur (based on the template used) and the level you want for this work item	no	12-char string: example: "Bx0xxxxxxxx9"	"Bx0123456789"		✓
maximum files not backed up before forcing full backup	Level 0 backup is forced for the filesystem work item when more than this number of files in the work item have never been backed up	no	number of files; 0 disables this feature	0		✓
Database Work Item Fields						
These "Database Work Items" pertain to the "offline database backup" functionality supported prior to EDM 4.4.0 and are included here to support restores only. They do not apply to the database backup clients currently supported for backup.						
work item	Name of the work item, the client backed up by the work item	yes	1-63 char (each, per work item & client name)	none		✓
filespec	Name of one or more client directories, filesystems, or files that receives level 0-9 backups via this work item	yes	pathnames and optional findxcpio qualifiers (no size limit)	none		✓
database type	Type of database to be shut down	yes	oracle, informix, sybase	none		✓

Table B-1

Summary of Fields in the Configuration File (Continued)

Field	Description	Req	Size or Values	Initial Setting	Takes Effect	
					Now	Next Bkup
database server	Name of the database server for the work item	yes	server name	none		✓
database name	Name of database(s) found during configuration	yes	database names	none		✓
type	Type of work item	yes	coordinated	coordinated		✓
use connection method	Defines the TCP/IP port on which the client is listening for a command from the EDM Backup server. Do not use for kicker work items	yes	socket@ <i>port</i>	socket@0		✓
backup client initialization command	Script to be run on the client before shutdown	yes	See "Database Work Item Fields" on page B-50	none		✓
backup client cleanup command	Script to be run on the client after restart	yes	See "Database Work Item Fields" on page B-50	none		✓
backup client data buffer size	Data buffer sizes on client and servers	yes	0-32 MB	0 MB		✓
backup server data buffer size		yes	0-32 MB	0 MB		✓
recovery client data buffer size		yes	0-32 MB	0 MB		✓
recovery server data buffer size		yes	0-32 MB	0 MB		✓

PC Work Item Fields

These fields are used for all PC clients: NetWare, Windows NT, and OS/2 and for OpenVMS Clients. In several cases the word *netware* is embedded in the code, but the field is used for all PC clients.

Table B-1

Summary of Fields in the Configuration File (Continued)

Field	Description	Req	Size or Values	Initial Setting	Takes Effect	
					Now	Next Bkup
work item	Name of the work item, the client backed up by the work item	yes	1-63 char (each, per work item & client name)	none		✓
filespec	Name of one or more client directories, filesystems, or files that receives level 0-9 backups via this work item	yes	pathnames and optional findxcpio qualifiers (no size limit)	none		✓
exclusion tag	ensures single-threaded processing for the client. Required and automatically generated for DOS.	no		none		✓
use connection method	Defines the TCP/IP port on which the client is listening for a command from the EDM Backup server	yes	netware@ <i>port</i> (see "Connection Method" on page B-57)	netware@1776 for NetWare netware@1492 for OS/2, netware@3895 for Windows NT netware@3896 for OpenVMS		✓
netware username	Determines what user privilege can execute backups and restores	yes	1-63 chars (for NT: max 20 characters)	none		✓
netware encrypted password	Encrypted password for netware username	yes	not editable directly. Enter from EDM Backup Configuration window	none		✓
netware client TSA	Assigns a target service agent to the PC server	yes	<i>servername</i> .FileSystem (.local for OS2)	none		✓

Table B-1

Summary of Fields in the Configuration File (Continued)

Field	Description	Req	Size or Values	Initial Setting	Takes Effect	
					Now	Next Bkup
netware client target	Defines the PC client as a target in need of service	yes	client name	none		✓
Backup Trailset (Media Set) Fields						
backup trailset	Name of the trailset	yes	1-63 char	primary		✓
use trail	Name and type of media used for the trail, level of backups written to the trail, and the maximum number of work items that can be backed up concurrently to this trail	yes	name: 1-11 char type: <i>media_type</i> level: 0-9, B1, B2 max clients: 1-24	none <i>media_type</i> none 8	✓ max client	✓ other fields
use level <i>n</i> for baseline backups	Level of baseline backup written to this trail set (applicable with backups of HSM systems that are scheduled automatically)	no	B1 or B2	none	next time scheduling occurs	
keep backups	Retention period for the backup media, qualified by level (0-9)	no	<i>nn</i> months or forever (<i>months</i> is seconds, days, weeks, months, or years)	1 yr (lev 0) 3 mos (lev 1-9)	✓	
keep backup catalogs	Retention period for backup catalogs, qualified by level (0-9)	no	same as for <i>keep backups</i>	1 yr (lev 0) 3 mos (lev 1-9)	✓	
keep saveset records	Retention period for saveset records, qualified by level (0-9, B1, B2)	no	same as for <i>keep backups</i>	1 yr (lev 0) 3 mos (lev 1-9)	✓	
backup catalog delta level	Backup level at which EDM Backup should consolidate catalogs	no	1-9	9	✓	

Table B-1

Summary of Fields in the Configuration File (Continued)

Field	Description	Req	Size or Values	Initial Setting	Takes Effect	
					Now	Next Bkup
duplicate media after backup on ..copies	Automatic media duplication	no	1	none		✓
manual activation of media duplication	Manual media duplication	no		none		✓
appending to current media copy	Append to current duplicate media for media duplication, as opposed to use new media.	no		none		✓
using new media at each duplication	Use new media for each duplication	no		none		✓
Backup Schedule Template Fields						
backup template	Name of the schedule template	yes	1-63 char	default		✓
work group list	Work groups that are backed up using this schedule template	yes	1-63 char (per work group)	none		✓
begin trailset rotations on	Date in future when EDM Backup should start using the trailset	yes	<i>dd-mmm-yy</i> <i>mm/dd/yy</i> <i>mmm dd, yy</i> <i>mmm dd, yyyy</i>	date template is added	next time scheduling occurs	
rotation period	Period of time during which each client should receive at least one level-0 backup	yes	<i>nn</i> days (<i>days</i> is days, weeks, months, or years)	14 days	next time scheduling occurs	

Table B-1

Summary of Fields in the Configuration File (Continued)

Field	Description	Req	Size or Values	Initial Setting	Takes Effect	
					Now	Next Bkup
primary trailset	Name of the trailset to which backups are written for all the work groups backed up using this template	yes	1-63 char	primary		✓
alternate trailset	Name of the trailset used to provide an optional, second set of backups on alternating nights (generally only used with automatic scheduling)	no	1-63 char	none		✓
logging level	Level of logging messages written to the file specified via the <i>server log file</i> field	yes	none errors stats debug per file	stats	✓	
server log file	Name and maximum size of the template's log file (stored on the server as /usr/epoch/EB/log/ <i>filename</i>)	no	pathname & file size size: <i>nnn</i> K or no limit (<i>K</i> is KB, MB, or GB)	default_ template.log 256KB	✓	
backup completion script	Name of the script file used to store or mail backup completion reports	no	pathname	mailok	✓	
backup failure script	Name of the script file used to store or mail backup failure reports	no	pathname	mailerr	✓	
do all baseline backups before normal backups	Statement used to force all scheduled baseline backups to finish before any level 0-9 backups start for the template	no	as stated	statement omitted (don't force baselines first)		✓

Table B-1

Summary of Fields in the Configuration File (Continued)

Field	Description	Req	Size or Values	Initial Setting	Takes Effect	
					Now	Next Bkup
recreate baseline if needed	Statement used to re-baseline files automatically, if the baseline copy is used in place of a staged-out copy during restore	no	as stated	statement omitted (do not recreate baseline)		✓
<i>schedule standard rotations</i>	Statement used to turn on automatic scheduling, and to schedule some portion of the clients for a full backup each day (cannot be specified with <i>full during weekends rotations</i>)	no	as stated	none	next time scheduling occurs	
<i>schedule full during weekends rotations</i>	Statement used to turn on automatic scheduling, and to schedule all full backups on Saturdays and Sundays (cannot be specified with <i>standard rotations</i>)	no	as stated	none	next time scheduling occurs	
weekday backup shift	Target amount of time EDM Backup should run each weekday (Monday-Friday; applicable with automatic scheduling to provide a guideline)	no	<i>hh</i> hours <i>mm</i> minutes in the range 1-24 hours	none		✓
weekend backup shift	Target amount of time EDM Backup should run each weekend day (Saturday-Sunday; applicable with automatic scheduling to provide a guideline)	no	<i>hh</i> hours <i>mm</i> minutes in the range 1-24 hours	none		✓

Table B-1

Summary of Fields in the Configuration File (Continued)

Field	Description	Req	Size or Values	Initial Setting	Takes Effect	
					Now	Next Bkup
[for <i>work group</i>] level <i>n</i> on ...	Statement(s) used to turn on custom scheduling, and to specify when to run the backups for each level, optionally qualified by work group	no	work grp: 1-63 char level: 0-9, B1, or B2 on: days (<i>days</i> is a number [1 or greater] within the rotation, as in 1, 3-7; or the actual days, as in Monday, Tuesday, 2nd Monday)	none	next time scheduling occurs	
Startup Parameters						
perform initial full backup on scheduled day	Statement used to perform the initial full backup on some portion of all newly installed clients each day during the first rotation period (applicable with automatic scheduling)	no	as stated	statement enabled	next time scheduling occurs	
perform initial full backup as soon as possible	Statement used to run the initial full backup on all newly-installed clients during the first backup run after those clients are installed (applicable with automatic scheduling, and suggested if all new clients should receive a full backup as soon as possible)	no	as stated	statement enabled	next time scheduling occurs	

Server Fields

The server fields apply to all backup and restore operations that the server performs. The server block looks similar to the following:

```
ebserver: "atlas1"
{
  client backup username: "ebadmin";
  backup administrator usernames: "root", "jan",
  "tracy";

  authorized backup list: "cad1", "cad2", "cad3",
  "cad4", "cad5", "cad6", "doc1", "doc2",
  "filserver1", "atlas1";

  authorized recovery list: "cad1:pat", "cad2:ken",
  "cad3:eric", "cad4:jane", "cad5:tom", "cad6:bob",
  "doc1:mary", "doc2:dave";

  authorized cross recovery list: "cad1:pat<cad4",
  "cad2:ken<cad6";

  recovery administrator list: "*:jane", "cad2:ken";

  maximum simultaneous clients: 4;
  use at most 2 dlt trails concurrently;
  limit throughput to: 600k per second;

  maximum server backups.log file size: 256k;
  maximum server recoveries.log file size: 256k;
  maximum client backups.log file size: 64k;
  maximum client recoveries.log file size: no limit;
  catalog threshold to force level 0: 0;
  .
  .
  .
}
```

The fields are described in order as shown above.

ebserver

This field specifies the name of the backup server, and defaults correctly when the server is configured at installation (with the **eb_server_config** command):

```
ebserver: "atlas1"
```

Client Backup Username

Use this field to specify the 1- to 64-character login name that **ebbackup** uses when it connects with client systems:

```
client backup username: "ebadmin";
```

Always use the installation default, **ebadmin**; this non-root user name is installed in `/etc/passwd` (or the NIS password map) for every client and server.

CAUTION: The ebadmin user name must have /bin/sh as its shell. Using any other shell causes installation and/or backup failures.

If the default conflicts with another login name on your network, use an account that is dedicated to doing backups. It must be common to the server and all clients, and it cannot be a regular user name.

If your site is running more than one server, specify the same login name in the configuration file for each server.

CAUTION: If you change the client backup username, you must reinstall the client software on every client, including the local client on the server.

Changes to this field take effect the next time **ebbackup** processing runs.

**Backup Administrator
Usernames**

Use this field to identify individuals who are responsible for EDM Backup:

```
backup administrator usernames: "root", "jan",  
"tracy";
```

You can specify any number of UNIX usernames (1 to 64 characters each, and set to "root" at installation). Each user whom you specify has root-like privileges *within the EDM Backup environment*.

The backup administrators can run backups and restore files from any client system, browse any backup catalog (regardless of restore permissions), perform cross-restores to restore any data to any client, and so on.

Note: Backup administrators do not necessarily need root access across the entire server system.

Repeat this field as many times as you want to add usernames. At a minimum, specify one name. Changes to this field take effect immediately.

Authorized Backup List

Use this list to identify those clients (workstations, file servers, or backup or migration server) whose files can be backed up by this backup server:

```
authorized backup list: "cad1", "cad2", "cad3",  
"cad4", "cad5", "cad6", "doc1", "doc2",  
"fileserver1", "atlas1";
```

Each 1- to 63-character client name must match:

- the client name that is entered in the NIS map or the local /etc/hosts file
 - the client name that is entered at installation. Make sure to use the expanded client name as registered in the NIS map (not an alias)
- and
- the client name that is specified in the work-item field

Repeat this field as many times as you want to add client names, or include all the backup clients in a single list. There is no default backup list. Unless you specify a client name here, it is not backed up.

Changes to this field take effect immediately.

Temporarily Disabling Backups for a Client

To disable backups temporarily for a client without changing its work-group definition(s), remove the client's name from this list. (You can still restore files backed up from the client.) When you put the client back in the list, backups resumes normally.

Authorized Recovery List

When you add a **Restore User Name** to the **Self Restore Permission** in the Backup Configuration window, that user name is added to the authorized recovery list in the eb.cfg file.

You can also add user names manually to the eb.cfg file. Use this field to identify users who can restore files that are backed up from their own client system, restoring them on their own (that is, the same) client:

```
authorized recovery list: "cad1:pat", "cad2:ken",  
"cad3:eric", "cad4:jane", "cad5:tom", "cad6:bob",  
"doc1:mary", "doc2:dave";
```

This ensures that only authorized users can perform a self-service restore. Unless you specify this field, no users can restore their own files.

Repeat this field as many times as you want to add user names, or include all of the specifications in a single list. Backup administrators can always restore their own files, so they should not be specified explicitly in this list.

Identify each item in the list as a client:user pair, in this format:

```
authorized recovery list: "client:user",  
..."client:user";
```

- Specify *client* as one of the following:
 - the name of the client system
 - an asterisk (*), which indicates any client

As an alternative to both the client and user names, specify the name of a netgroup preceded by an @, where the netgroup identifies each client:user combination that can perform a self-service restore:

```
authorized recovery list: "@doc";
```

- Specify *user* as one of the following:
 - the login name of the individual on that client who has permission to restore files to that client (illustrated below for cad1-cad3):

```
authorized recovery list: "cad1:*, "cad2:*",  
"cad3:eric";
```

- an asterisk (*), which means that any user who is logged into this client can restore the client's files:

```
authorized recovery list: "cad6:*";
```

as an equivalent to the asterisk, omit the colon and username completely:

```
authorized recovery list: "cad6";
```

Changes to this field take effect the next time a restore runs.

You can also disable self-service restores for all clients by leaving the list blank (or by omitting the list):

```
authorized recovery list: ;
```

Authorized Cross-Recovery List

Use this field to identify users who can restore files to client systems other than those from which the backups originated:

```
authorized cross recovery list: "cad1:pat<cad4",  
"cad2:ken<cad6";
```

Users who request a cross-restore must either own the files that are being restored, or must have *read* access to those files under UNIX. Users must also have write and execute permissions to the destination directory.

Repeat this field as many times as you want to add user names, or include all the specifications in a single list. Do not include backup administrators in this list. Backup administrators can always restore from any client system to any other client system, regardless of the system on which they are working.

Specify the list using the format shown below:

```
authorized cross recovery list: "client:user <
backup_client";
```

Note: The user requesting a cross-restore becomes the owner of the restored files, regardless of the client from which they were originally backed up. An exception occurs if the user requesting the cross-restore is also a restore administrator, in which case the original owner is preserved.

- Specify *client* as the system to which files can be restored through a cross-restore. At restore time, this must be the client at which the person requests the cross-restore is logged in. Specify:
 - the name of the client system to which files can be restored
 - an asterisk (*), which indicates any backup client (illustrated below) where Pat can restore any files to (and from) any client system):

```
authorized cross recovery list: "*:pat < *";
```

As an alternative to both the client and user names, specify the name of a netgroup preceded by an @, where the netgroup identifies each user that has permission to restore files to the clients in the netgroup:

```
authorized cross recovery list:"@netgroupa< cad5";
```

- Specify *user* as the name of the user authorized to initiate restore on the client identified to the left of the colon. You can specify:

- the login name of the individual who has permission to restore files to that client, as it appears in the authorized recovery list. The following example gives the user Pat permission to restore files that are backed up from cad6 to cad1:

```
authorized cross recovery list: "cad1:pat < cad6";
```

- an asterisk (*), which means that any user who is logged into the client can perform the cross-restore:

```
authorized cross recovery list: "cad1:* < cad6";
```

as an equivalent to the asterisk, omit the colon and username completely:

```
authorized cross recovery list: "cad1 < cad6";
```

- Specify *backup_client* as one of the following:
 - the name of a client from which files were originally backed up (where the files can be restored to the indicated client:user)
 - an asterisk (*), which indicates any backup client
 - the name of a netgroup that is preceded by an @, to specify the clients in an entire netgroup

The following example allows any user who is logged in to cad1 to restore files from any backup client:

```
authorized cross recovery list: "cad1 < *";
```

Changes to this field take effect the next time a restore runs.

Disabling All Cross-Restores

Select **Restrict Cross-Restore to Clients** in the Client tab in the EDM Backup Configuration window to disable all cross-restores, except those that the restore administrator list explicitly allows, and those that the backup administrator performs.

To do this manually in the eb.cfg file, leave this list blank (or omit the list) to disable all cross-restores:

```
authorized cross recovery list: ;
```

Recovery Administrator List

Use this field to identify users who can restore files other than their own to the client from which the files were originally backed up:

```
recovery administrator list: "*:jane", "cad2:ken";
```

The user who requests the restore must be logged in to the client from which the files were backed up.

This feature is designed for use when multiple users share a client, to enable a single user on that client to serve as the restore administrator for just that client. Repeat this field as many times as you want to add usernames, or include all the specifications in a single list. Backup administrators have all of the rights of a restore administrator, so should not be specified explicitly in this list.

Identify each item in the list as a *client:user* pair, in this format:

```
recovery administrator list: "client:user",  
..."client:user";
```

Note: The original owner is preserved for all files that a restore administrator restores.

- Specify *client* as one of the following:
 - the name of the client. This is the client from which the files were backed up and to which they can be restored by the restore administrator.
 - an asterisk (*) to indicate any client:

```
recovery administrator list: "*:karen";
```
 - the name of a netgroup preceded by an @, where the netgroup identifies each client in the netgroup.
- Specify *user* as the name of the restore administrator authorized to initiate restore on the client(s) indicated. You can specify:
 - the login name of the user.
 - an asterisk (*) to indicate any user:

```
recovery administrator list: "cad1:*";
```

As an equivalent to the asterisk, omit the colon and username completely:

```
recovery administrator list: "cadl";
```

Changes to this field take effect the next time a restore runs.

Disabling Administrator-Level Restores

Leave this list blank (or omit the list) to disable all administrator-level restore functions except those that the authorized cross-recovery list explicitly allows, and those that the backup administrator performs:

```
recovery administrator list: ;
```

Maximum Simultaneous Clients

Use this field to specify the *global* maximum number of work items (not clients) the EDM Backup server can back up concurrently, *across all trails* being written at any one time. This field controls the total amount of resources the server can allocate to backup functions, and relates to the server's available processing power, memory, and connectivity. Adjust this field to reduce or expand the system resources available for backup processing:

```
maximum simultaneous clients: 24;
```

Specify a value in the range 1-64 (set to 24 at installation). If you specify a value that is too small, EDM Backup under-utilizes system resources. If you specify a value that is too large, EDM Backup saturates the server's virtual memory, which causes memory thrashing and reduces performance.

Changes to this field take effect immediately.

Use At Most *n media-type* Trails Concurrently

Your server only has a fixed number of physical drives for backup. Besides EDM Backup, other applications on the server may need to use these drives, such as the HSM server.

Specify one *use at most* field for each type of backup media, to define the maximum number of drives EDM Backup can use for each device type (that is, the maximum number of trails EDM Backup can write to that type of media at once). This is initially set to 4 for each type of device that is available to the server.

This field has the format:

`use at most n media-type trails concurrently;`

where:

- *n* is an integer representing the number of server drives EDM Backup can use at any one time.
- *media-type* indicates the type of drive.

For example, if the EDM Backup server has four tape drives, but wants to reserve two drives for non-backup purposes, you specify:

`use at most 2 dlt trails concurrently;`

Changes to this field take effect immediately, although no backups are terminated to comply with the new limits.

Limit Throughput To: *nnn* Per time

Use this field to limit the amount of the network bandwidth available to EDM Backup, thereby allowing the network to accommodate applications other than EDM Backup. With this field specified, EDM Backup monitors its network use. If it reaches the specified limit, it does not start another client backup until the throughput drops. (However, EDM Backup does not stop any backups currently in process if the throughput exceeds this limit.) Specify *no limit* if you don't want EDM Backup to monitor the network.

Specify this field in terms of the number of bytes EDM Backup can use during a specific time period:

`limit throughput to: bytes per time-units;`

Where:

- *bytes* is the number of bytes to which you want to limit the throughput, followed by a unit code (set to 600KB per second at installation). Specify the unit code as follows:

Unit Code	Measure
k, K, kb, or KB	kilobytes
m, M, mb, or MB	megabytes
g, G, gb, or GB	gigabytes

- *time-units* defines the unit of time during which you want to limit throughput to the number of bytes specified:

Time Code	Limits Throughput Based on the Specified Number of Bytes Per:
second(s)	second
minute(s)	minute
hour(s)	hour

You can combine the time units, as in this example:

```
limit throughput to: 15MB per 1 hour 30
minutes;
```

Here are some guidelines:

- If the server is connected to a single Ethernet, set this to 6 (600KB/second).
- If you have high end clients, two independent FDDI rings, 6 CPU SC-1000, and 8 DLT drives, you can set this for up to 200 (20MB/second).

Changes to this field take effect immediately.

Specifying No Throughput Limit

If you want backups to proceed as quickly as possible with no restrictions on throughput, specify *no limit* (and no other options). You may want to do this if you have FDDI or multiple Ethernets.

```
limit throughput to: no limit;
```

Maximum Server backups.log File Size

Use this field to specify the maximum size of the backup log file (/usr/epoch/EB/log/backups.log) on the server (generally in the range 16KB-256KB). This file provides a record of all backup activities, limited only according to the size restriction that is specified here. Determining this size is a trade-off between using disk space on the server versus keeping the backup history for a longer period of time.

When the file reaches the specified size, EDM Backup locates the oldest data in the file and expires ten percent of that data to free up space for new information.

The format of this field is:

```
maximum server backups.log file size: file-size;
```

Where *file-size* indicates how large the file can get before the oldest data is expired. Specify this field as a number of bytes followed by a unit code (set to 256KB at installation). Specify the unit code as follows:

Unit Code	Measure
k, K, kb, or KB	Kilobytes
m, M, mb, or MB	Megabytes
g, G, gb, or GB	Gigabytes

If you want to let the file grow until it is as large as the physical storage space allows, specify *no limit*. By not setting a limit on the file, it becomes a permanent record of backup operations:

```
maximum server backups.log file size: no limit;
```

This example sets the maximum file size for backups.log to 200KB:

```
maximum server backups.log file size: 200KB;
```

Note: If you do not limit the file size, it may become too large to manage. Also, in HSM systems it is not staged.

Changes to size of the backup log file take effect immediately.

Maximum Server recoveries.log File Size

Use this field to specify the maximum size of the restore log file (/usr/epoch/EB/log/recoveries.log) on the server (set to 256KB at installation). This file provides an audit trail of all restore activities, limited only according to the size restriction specified here. It can be used to examine exactly what occurred during restore processing. Determining this size is a trade-off between using disk space on the server versus keeping audit trails on the server for a longer period of time.

The format of this field is:

```
maximum server recoveries.log file size: file-size;
```

For example:

```
maximum server recoveries.log file size: 200KB;
```

Specify this field exactly as described for the server's backup log file. With the exception of when changes take effect, these two settings are handled exactly the same.

Changes to the size of the restore log file take effect the next time a restore runs.

Maximum Client backups.log File Size

Use this field to specify the maximum size of the backup log file (~ebadmin/*client-name*/backups.log) on each client (set to 64KB at installation). This file provides a brief record of backup activities for the client. The format of this field is as follows:

`maximum client backups.log file size: file-size;`

For example:

`maximum client backups.log file size: 64KB;`

Specify this field exactly as is described for the server's backup log file. With the exception of the default and when changes take effect, these two settings are handled exactly the same.

Changes to this field take effect the next time the corresponding client is backed up.

Maximum Client recoveries.log File Size

Use this field to specify the maximum size of the restore log file (`~ebadmin/client-name/recoveries.log`) on each client (set to 64KB at installation). This file provides a brief record of restore activities for the client, and can be used to locate the more comprehensive restore information on the server.

The format of this field is:

`maximum client recoveries.log file size: file-size;`

For example:

`maximum client recoveries.log file size: 32KB;`

Specify this field exactly as is described for the server's backup log file. With the exception of the default and when changes take effect, these two settings are handled exactly the same.

Changes to this field take effect the next time the corresponding client is backed up.

Catalog Threshold to Force Level 0 Backup

This statement directs catalog processing to schedule a full (level 0) backup (instead of an incremental) for any filesystem work item when it detects too many files within that work item that have never been backed up. (Applies to filesystem backups only, as database backups are always full.)

The concern is for files that do not get backed up during an incremental backup because the files were added in a manner that preserved the creation and access time of the file prior to the last backup of the work item. In this case, you want the next backup for that work item to be a level 0. Failure to do this causes catalog processing to take a long time and could result in a large number of files not being backed up, meaning they could not be recovered.

Also, see “When You Change a Work Item or a Filesystem” on page B-35 for times you should force a level 0 backup.

The format of the field is as follows:

```
catalog threshold to force level 0 backup:  
threshold;
```

For example:

```
catalog threshold to force level 0 backup: 10;
```

If **ebcatcomp** detects more than *threshold* files that have never been backed up for a filesystem work item, it schedules (command line scheduling) a level 0 for that work item. If *threshold* is set to zero, the default value, this feature is disabled.

Note: This statement can increase the number of level 0 backups performed.

The following alternate wording for this statement is acceptable in the server block:

```
maximum files not backed up before forcing full  
backup: value;
```

This statement can be overridden by a version of this directive that can be set within individual filesystem work items. See “Maximum Files not Backed Up Before Forcing Full Backup” on page B-49.

Work Group Fields

Work groups define a set of work items of the same type so that a template can back them up together (during the same shift and using the same trailset) just by referencing their work-group name. Specify as many work groups as necessary to configure your site. If all your clients can be backed up together by using a single trailset, specify one work group that includes all client systems.

The work-group block looks similar to the following:

```
work group: "doc"
{
  work-item definitions ...
}
```

Note: A work group can be referenced by more than one template. While this may result in backing up the clients in the work group more often than necessary, it does not cause any conflicts (in media, scheduling, etc.).

Work items in a work group must be of the same type. For example, a work group cannot contain file system work items with NetWare work items or database work items.

Specify the work-group name as described below, then proceed to the discussion on coding work items.

Work Group Name

Specify the work-group field as any unique 1-63 character name used to reference a group of clients, in the following format:

```
work group: "work-group name"
```

Note: The work-group name field does not end with a semicolon, but is followed by a brace-delimited block that defines its work items.

Changes to the work-group name take effect the next time **ebbackup** processing runs.

Filesystem Work Item Fields

Each work group is comprised of one or more *work items*. A file system work item defines a set of files to be backed up for a single client, and information relating to those files (whether to back up all the files, resident files only, and so forth). Each client requires at least one work item before it can be backed up by EDM Backup.

The following sample shows a work group. (The migration, baseline, and completeness fields are used with HSM.) The *local* work group backs up files on the EDM Backup server (named atlas1), and includes three work items:

- one work item backs up key database files

- two work items back up the non-database files from each of two filesystems (/ and /home).

```

work group: "local"
{
  {
    work item: "atlas1:LOCAL_DATABASE", "atlas1"
    {
      filespec: LOCAL_DATABASE;
      completeness: DB_COMPLETENESS;
      priority: PRIORITY_SERVERDB;
      level map: LEVEL_MAP_SERVERDB;
    }
  }
  {
    work item: "atlas1:/", "atlas1"
    {
      filespec: "/" -xdev";
      exclusion tag: "/dev/rsd2C/c0t3d0s4";
      baseline filespec: "/" -xdev -staged";
      completeness: non-baselined files only;
    }
  }
  {
    work item: "atlas1:/home", "atlas1"
    {
      filespec: "/home -xdev";
      migration backup tag: "local_home"
      exclusion tag: "/dev/rsd2C/c0t3d0s4";
      baseline filespec: "/home -xdev -staged";
      do not load balance;
      completeness: non-baselined files only;
    }
  }
}

```

Work item used to back up server database files

Work item used to back up the server's root (/) filesystem

Work item used to back up the server's /home filesystem

Some of these options only apply for a particular type of EDM Backup client, as detailed in Table B-2.

Table B-2

Work Item Options Available by Client Type

Option	Applicable if the work item is:			
	on a Migration Client	on a Non- Migration Client	Local to the Backup Server	
			Level 0-9	Level B1 & B2
work item:	yes	yes	yes	yes
filespec:	yes	yes	yes	—
baseline filespec:	—	—	—	yes
migration backup tag:	—	—	yes ¹	—
exclusion tag:	yes	yes	yes	yes
priority:	yes	yes	yes	yes
do not load balance;	yes	yes	yes	—
completeness:				
<i>all files</i>	— ²	yes ⁴	yes	—
<i>resident files only</i>	yes ³	—	yes	—
<i>files not backed up in migration store</i>	yes ⁴	—	—	—
<i>non-baselined files only</i>	—	—	yes ⁴	—
level map:	yes	yes	yes	yes

1. Applies when backing up staged files on an EDM Backup server that is also an EDM Migration server.

2. Only used when you reach a project milestone or temporarily decommission EDM Backup, to record the filesystems exactly as they stand at that time.

3. This setting can leave you vulnerable unless there's a backup of the client store. Specifically, if you don't back up a file that's been staged out, but you lose the file's client store on the server before the server's files are backed up, the only way you'll be able to restore the file is from an old backup.

4. Default for this client type.

Specify each work-item field as described in the following discussions. The fields are covered in order as shown in the syntax above. Repeat the *work-item* specifications as many times as necessary to configure backups for each client.

All changes that are made to the work item take effect the next time **ebbackup** processing runs.

When You Change a Work Item or a Filesystem

When you change a work item's file specification significantly (for example, to add a new filesystem to the backup list), always schedule the next backup for that work item as level 0, and make sure it is backed up the next time EDM Backup runs. Failure to do this causes catalog processing to take a long time and may result in a large number of files not being backed up.

Other times to force a level 0 are:

- When a filesystem covered by a work item changes its device number (such as when reformatted or when moved to a new system with the same hostname).
- When you have added files to a filesystem covered by a work item in a manner that preserved the creation and access time of the file prior to the last backup of the work item. See "Catalog Threshold to Force Level 0 Backup" on page B-29 and "Maximum Files not Backed Up Before Forcing Full Backup" on page B-49.

When You Stop Using a Work Item


If you stop using a work item — for example, after you create two work items for a client that previously had one — make sure to keep the old work item in the configuration file. If you delete that older work item, you lose its association with its client system, resulting in problems when you run history reports.

To avoid this situation, define one work group that contains only obsolete work items, and that is not backed up by any template. This retains the association between each old work item and its client, for each work item in the group.

Work Item Name

Specify a 1-63 character name that is unique within the configuration file. As a convention, it is good to incorporate the name of the client when specifying the work-item name, as well as some indication of what files are backed up using the work item. EDM Backup generates work-item names that look similar to the following when each client is installed (shown for a work item that backs up *all* files on the *doc1* client):

```
work item: "doc1-all", "doc1"
```



At a minimum, there is one work item per disk; there can be several work items per client.

Note: In a HSM system, when defining local-client work items, always specify one work item to correspond to each client store that migrated to the server. Use that work item to back up the client store on the server. Make sure to include a Backup/HSM tag in the work-item definition, and reference the same tag in the corresponding store-name definition.

A work item can back up an unlimited number of files. However, a work item that backs up more than 250,000 files will have substantially slower catalog processing.

This example shows the *doc1* client broken down into three work items:

```
work item: "doc1o/", "doc1"
{
    filespec: " ... ";
}
work item: "doc1o/usr", "doc1"
{
    filespec: " ... ";
}
work item: "doc1o/other", "doc1"
{
    filespec: " ... ";
}
```

Client Name

Specify the name of the client to which the work item applies, as it appears in the authorized backup list. Always use the primary complete name for the client; not an alias.

```
work item: "doc1-all", "doc1"
```

Client Name



Filespec to Back Up

This field lists each directory, filesystem, and/or file that you want to back up using the work item. This field has a 4096 character limit.

Specify the exact names as they appear in a directory listing by using a syntax similar to **find**. For each directory or filesystem, include the name and optional qualifiers. Refer to Appendix D “findxcpio Directives” for more details about the syntax and the semantics of specifications that this field supports. (Also see the **findxcpio** man page.)

Using the Block Form of the Syntax (Filespec Statement)

The work-item syntax uses a separate statement (not on the same line as the work-item name).

```
work item: "work-item name", "client name"
{
  → filespec: "filespec to back up for level 0-9 backups";
    baseline filespec: "filespec to back up when baselining";
    migration backup tag: "migration backup template name";
    exclusion tag: "tag";
    priority: priority;
    do not load balance;
    completeness: completeness-code;
    level map: "level map";
}
```

Backing Up the Local Client

It is important that you back up all filesystems that are stored on your backup server. The server stores the database files for all EMC backup and HSM products, as well as the bitfiles (client stores) written out by EDM Migration. Also, the product installations modify several standard UNIX files, such as `/etc/passwd`, `/etc/group`, and `crontab`.

There is a special `#define` macro for use with the local client, which causes it to back up all the server's critical database files:

```
filespec: LOCAL_DATABASE;
```

The following macro is included in the autoconfigured work item for the server's database files (shown below for the `atlas1` server), and should not be changed:

```
work item: "atlas1:LOCAL_DATABASE", "atlas1"
{
    filespec: LOCAL_DATABASE;
    completeness: DB_COMPLETENESS;
    priority: PRIORITY_SERVERDB;
    level map: LEVEL_MAP_SERVERDB;
}
```

Baseline Filespec

The baseline filespec is available if you purchased HSM, and applies when you back up the EDM Backup server's own files — generally only the files that are used for stageable filesystems. It lists each filesystem for which you want to maintain a baseline backup (level B1, B2, or both), in this format:

```
baseline filespec: "filespec to back up when baselining";
```

If you want to maintain a baseline backup for `/home`, you specify:

```
baseline filespec: "/home -xdev -staged";
```

Migration Backup Tag

Include a migration backup tag for use with level 0-9 backups on a local client, when:

1. The EDM Backup server is also an EDM Migration server and
2. The files that are being backed up are the staged (migrated) versions of client files

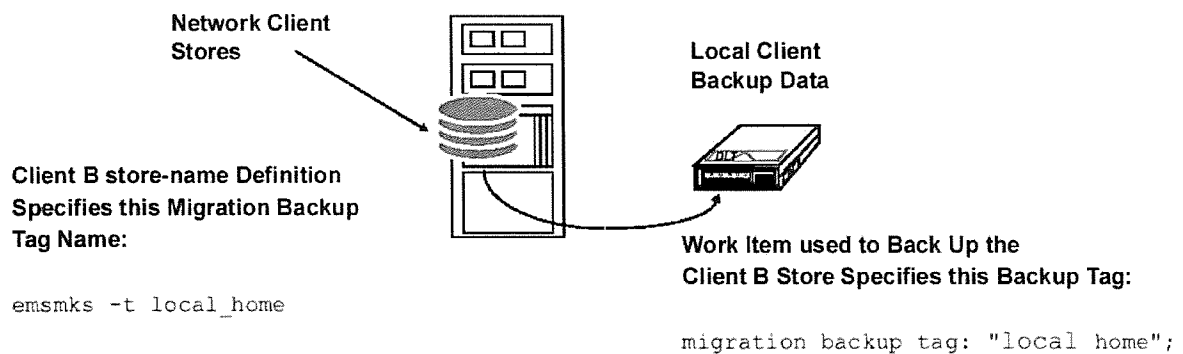
A migration backup tag is required in this case. Specify one work item for each client store, and include this field:

migration backup tag: "*migration backup template name*";

Where *migration backup template name* is a 1-63 character identifier. The first 16 characters must be unique across all work items that back up the server's own files, and must match *exactly* the migration *backup template name* that is specified in the definition of the migration store used to stage the same files from their network client.

Figure B-1

Migration Backup Tag



The migration backup tag is used to update the backupdates database (/usr/epoch/etc/backupdates), so that a subsequent backup of a network client that is migrated to this server can determine whether the staged-out client files were already backed up on their migration server (that is, the server where you define the work item).

Exclusion Tag

Use this field to identify any work item that should not be backed up concurrently with one or more other work items. For each client, specify the same tag value (1-63 characters) for all those work items that are mutually exclusive, using this format:

```
exclusion tag: "tag";
```

The combination of the exclusion tag and client name forms a unique identifier within the configuration file, but you can use the same tag value for any number of clients.

Typically, you use an exclusion tag for work items that back up data stored on the same physical disk, to prevent disk thrashing during backup processing (shown below for two work items that back up files from the rsd2C/c0t3d0s4 device):

```
work item: "atlas1:/", "atlas1",
{
    filespec: "/" -xdev";
    exclusion tag: "/dev/rsd2C/c0t3d0s4";
.
.
work item: "atlas1:/home", "atlas1",
{
    filespec: "/home -xdev";
    exclusion tag: "/dev/rsd2C/c0t3d0s4";
.
.
.
```

If no exclusion tag is included for a work item, **ebbackup** assumes it can process that work item concurrently with any other work item.

Connection Via

You can install just one copy of the EDM client software on UNIX client machines that are equipped with more than one Ethernet, FDDI, Token-Ring, or ATM port and use it to have your backups obtain higher throughput rates by taking advantage of the multiple network ports.

Setting up the Network

If possible, at least one of the network cards for the EDM and the clients should be connected to a separate network dedicated to backups. This reduces the amount of network traffic on the customers' main network and reduces the impact of backups on the users.

The EDM should be configured on the network in a way that each card uses a single/different client port. That is, card "A" on the server should talk to "A" on the client and card "B" on the server should talk to "B" on the client. Each client port must have a distinct clientname. For example: clientname_A.

Configuring the Work Items

You can configure the client's backups to be divided into separate backup streams (each defined by a "work item"). You can specify (in the work item) an alternative port name for one or more of the backup streams. The clientname is used as the name of the default port for backups. (It can match or differ from the hostname of the client machine.)

To set up a filesystem work item with an alternate network port, add the distinct clientname to the EDM Backup Configuration window: select **Work Item Options** and enter it in the **Generic** tab. To set up online database backups, use the **Configure Online Options** in the Client tab.

This puts the clientname in a **connection via** parameter in the work item statement in eb.cfg, which specifies it as the alternate port name.

Priority

Use this field to assign a priority to the work item, forcing it to run either before or after one or more other work items.

When you start a backup, **ebbackup** searches through its schedule for the work items that have the highest (that is, lowest-numbered) priority. It makes that priority *current*, and processes all work items that have that current priority to

completion. Then it searches for the next lower priority and makes it current, and so forth, until all work items are processed.

Note: The use of priorities limits the flexibility that is available to EDM Backup in choosing what backups to run next, and might reduce the efficiency with which the server's resources are used.

If a work item is added to the schedule and has a priority that's higher (i.e., numerically lower) than the one being processed, that work item is started as soon as server resources are available, and its priority becomes the current priority. This might be the case, for example, if a request is submitted online, or **cron** starts a scheduled backup while another backup is still running.

Use this syntax to specify the priority:

`priority: priority;`

Where *priority* takes one of two forms:

- assigned integer in the range -25 to 50, where -25 specifies the highest priority (run first) and 50 specifies the lowest priority (run last).
- one of the following #define macros:

Table B-3

Priority Settings

Macro	Equivalent	Description
PRIORITY_FIRST	-25	Backups should run before the general backups.
PRIORITY_DEFAULT	0	Backups should be considered part of the general backups (no special priority). This is the initial setting at installation, except for the work item used to back up the server's database files.
PRIORITY_LAST	25	Backups should run after the general backups.
PRIORITY_SERVERDB	50	Backups should run last. This keyword is reserved for use in backing up the server's database files (LOCAL_DATABASE).

If the backup server is also an HSM server, EDM Backup automatically backs up the stores for each migration client before backing up the client files. You do not have to be concerned with setting a higher priority for the local-client work items in this case. That way, if the level 0-9 backup of a networked migration client calls for a backup of a migrated file, and the work item specifies a completeness option of *files not backed up in migration store*, the file do not have to be copied back in and backed up again.

To ensure that the migration stores are backed up first for each migration client, you must:

1. Use the same priority for the network client and all work items that back up the client stores on the server (generally priority 0, the default).
2. Set the migration tags correctly for all work items that back up the client stores on the server.

Note: If you purchased HSM (with baselining) and the template specifies *do all baseline backups before normal backups*, the baseline backups always run before any level 0-9 backups for that template, regardless of the priority for any specific work item.

Do Not Load Balance

Load balancing works in conjunction with automatic scheduling, and is the method whereby EDM Backup schedules backups as evenly as possible across all clients. By default, EDM Backup assumes that each work item participates in load balancing, which can result in some work items having more than one level 0 backup during any given rotation.

Include the following statement if you do not want the work item to participate in load balancing:

```
do not load balance;
```

This ensures that the work item does not receive more level 0 backups during the rotation period than are called for. This might be useful, for example, for filesystems that take a long time to back up, or for production systems whose resource consumption is critical.

Completeness

Use this option for files under migration control, to avoid performing duplicate backups of the same file data:

Note: You can use a special #define completeness macro, DB_COMPLETENESS, with the local client. This macro equates to *all files*. It is included in the autoconfigured work item for Epoch's database files and should not be changed.

Use this option for files under migration control, to avoid performing unnecessary backup. It limits the files for which the *data portion* is written to the backup. (Remember that the *extended inode* is included for each file scanned, regardless of whether its data is written out.)

`completeness: completeness-code;`

Specify the *completeness-code* as any of the values below. This field applies for level 0-9 backups. The initial setting varies depending on the type of file that is being backed up (as noted in Table B-4). Generally it is a good idea to use the defaults.

Table B-4**Completeness Settings in HSM Systems**

Code	Description	Applicable For	Default For
All files	Back up the data portion of all files in the filespec, regardless of where they're stored and whether they're baselined.	All clients (this is the only option available for backup clients that are not also EDM Migration clients)	Non-migration clients & server's database files on the server
Resident files only ¹	Back up the data portion for only those files that are resident (local to) the client; or, for the server, that are stored on the magnetic disk.	Levels 0-9 on the backup server (i.e., the local client), and EDM Migration clients	—
Files not backed up in migration store	If a file has been staged, only back up the data portion of the file if its staged version hasn't been backed up yet on the HSM server.	EDM Migration clients	EDM Migration clients
Non-baselined files only	Back up the data portion for only those files that aren't baselined (for use after a baseline is taken). This option is only available if you've purchased HSM.	Local (server) client (but not server's database files)	Local client (except server's database files)

1. It's safest to use the next option (files not backed up in migration store) with EDM Migration network clients (and non-baselined files only for the local EDM Migration client). The resident files only setting can leave you vulnerable unless there's a backup of the client store. Specifically, if you don't back up a file that's been staged out, but you lose the file's client store on the server before the server's files are backed up, the only way you'll be able to restore the file is from an old backup.

Level Map

Use this option to override the backup level(s) that normally occur for the work item, based on the template that is used. This feature is useful when a few work items need a level of backup that is different from most of the work items referenced by a template, and saves having to define a separate template for those work items.

The level that a level map specifies completely replaces (in the schedule) the level that is otherwise performed. EDM Backup does not keep a record of the old (replaced) level.

Note: The level map applies for custom and automatic scheduling. Command-line scheduling does not use the level map.

Specify the level map by using one of the #define macros that are listed in the following table. For example:

```
level map: LEVEL_MAP_SKIP_ALL_BUT_L0;
```

Table B-5

Level Map Settings

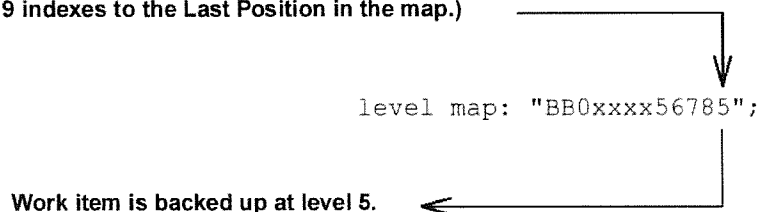
Level Map #define Macro	Equivalent Value	Description
LEVEL_MAP_SKIP_ALL_BUT_L0	"BB0xxxxxxxxx"	Only run level-0 and baseline backups. Skip all other levels.
LEVEL_MAP_EVERYTHING_IS_L0	"BB0000000000"	Run all scheduled backups as level-0, regardless of the level specified by the template.
LEVEL_MAP_SKIP_ALL	"xxxxxxxxxxxx"	Skip the work item completely. Don't run any backups.
LEVEL_MAP_SERVERDB ¹	"BB0000000000"	Same as LEVEL_MAP_EVERYTHING_IS_L0, but reserved for when backing up the EDM Backup server's database files.

1. The LOCAL-DATABASE work item (for EDM Backup database files) must always receive a level 0 backup, using the LEVEL_MAP_SERVERDB level map. This simplifies restoring those critical files.

Alternatively, specify any 12-character string you want, where each character is indexed to a backup level (B1, B2, "BB" then 0-9, in that order). Use an *x* to suppress a level of backup

completely. Any other value refers to the level of backup that should occur when the work item is processed at the indexed level.

Template requests a Level 9 Backup.
(Level 9 indexes to the Last Position in the map.)



The following string requests that each backup proceed at the normally scheduled level, and indicates that there is only one baseline backup:

```
level map: "Bx0123456789";
```

Though you cannot remap a baseline backup to a different level, you can cause it to be ignored. If the trailset requests a B1 backup, for example, you cannot change it from the first id (level map: "Bx01...") to the second id (level map: "xB01..."), but you can suppress it entirely:

```
level map: "xx0123456789";
```

Here are some examples of level mapping:

- To change level 6-9 backups to a level 5 for the work item:

```
level map: "BB0123455555";
```
- For stable filesystems that seldom change, run the baseline and level-0 backups as scheduled, but ignore other levels:

```
level map: "BB0xxxxxxxxx";
```
- For filesystems that change frequently but for which incremental backups are not used during restore (as for the EDM Backup databases), run all backups as level 0:

```
level map: "BB0000000000";
```

- If you are running level 9 backups manually during the day (in addition to the automatically scheduled nightly backups), you might map the nightly level 9 backups to level 5, then specify a separate trail (set of media) for the levels 5 and 9. The trail that is used to restore files do not involve the manually run level 9 incrementals that are taken throughout the day:

```
level map: "BB0xxxx5xxx5";
```

Because the level map is not used when you run command-line backups, the level 9 incrementals that are run from the command line actually perform level 9 backups.

- If you want to suppress all backups:
level map: "xxxxxxxxxxxx";

Access Time Preservation

Two alternative statements are available for clients that do not allow *invisible file access*.

(With *invisible file access*; neither atime nor ctime is updated when files are backed up. Clients that support invisible file access: Solaris 2.1 or higher for SPARC, SunOS 4.1.3 for SPARC. This setting does not apply to these clients.)

For all other clients, you can decide whether each night's backup process itself either:

- preserves the original change time (ctime), but updates each file's access time (atime) to the time of the backup, or
- preserves the original access time (atime), but updates each file's change time (ctime) to the time of the backup

To preserve the original ctime, use
`preserve file change time during backup;`

To preserve the original atime, use
`reset file access time after reading file;`

Note: With “reset file access time after reading file;” incremental backups do not check the ctime for changes as they do with “preserve file change times during backup;”. The result: for applicable platforms, if you perform functions that change the inode, such as **chmod**, these changes are not recorded during incremental backups.

These two options are mutually exclusive. (If both are applied, a semantic warning is issued during backup processing and then the system’s default operation — preserving ctime but updating atime — is performed.)

Maximum Files not Backed Up Before Forcing Full Backup

This statement directs catalog processing to schedule a full (level 0) backup (instead of an incremental) for this filesystem work item when it detects too many files within this work item that have never been backed up. (Applies to filesystem backups only, as database backups are always full.)

The concern is for files that do not get backed up during an incremental backup because the files were added in a manner that preserved the creation and access time of the file prior to the last backup of the work item. In this case, you want the next backup for that work item to be a level 0. Failure to do this causes catalog processing to take a long time and could result in a large number of files not being backed up, meaning they could not be recovered.

Also, see “When You Change a Work Item or a Filesystem” on page B-35 for times you should force a level 0 backup.

The format of the field is as follows:

```
maximum files not backed up before forcing full  
backup: value;
```

For example:

```
maximum files not backed up before forcing full  
backup: 10;
```

If **ebcatcomp** detects more than *value* files that have never been backed up for this filesystem work item, it schedules (command line scheduling) a level 0 for this work item. If *value* is set to zero, the default value, this feature is disabled.

Note: This statement can increase the number of level 0 backups performed.

This statement overrides a version of this directive that can be set in the server block, which applies to any filesystem work item. See “Catalog Threshold to Force Level 0 Backup” on page B-29.

Database Work Item Fields

Note: These “Database Work Items” pertain to the so-called “offline database backup” functionality supported prior to EDM 4.4.0 and are included here to support restores only. These work item specifications do not apply to the various database backup clients that are currently supported for backup.

Support for new backups using the “offline database backup” functionality are not supported as of EDM 4.4.0. Nor does EDM 4.4.0 support any reconfiguration of these database work items or configuration of new ones.

However, support for restore of backups taken using the “offline database backup” functionality continues in EDM 4.4.0, and to be able to restore such a backup depends on the continued presence of its particular database work item in your eb.cfg. Hence this section continues to be included in this revision of the *EDM Software Reference* manual.

Note: The work items that pertain to the various database clients are documented in the individual database client guides as appropriate; they are not included in this appendix.

A database work item defines a set of databases to be backed up. The following is an example.

```
work item: "chipmunk:master+:sybase", "chipmunk"

{
  filespec:"DO_FILE_LIST /ext_ibm/sybase/master.dat";
  database type: "sybase";
  database server: "SYBCHIP";
  database name: "master model tempdb";
  inclusion tag: "chipmunk:SYBCHIP";
  type: coordinated;
  backup client initialization command: "-exec_as
    'sybase' -DBServer 'SYBCHIP' ebcv_shutdown
    sybase";
  backup client cleanup command: "-exec_as 'sybase'
    DBServer 'SYBCHIP' ebcv_startup sybase";
  use connection method: netware@514;
}
```

In the discussions that follow, the fields are covered in order as shown in the example above. The *work item* specifications are repeated as many times as necessary to configure backups for each client.

CAUTION: These database work items must remain as they are for restores of their corresponding backups to work.

Database Work Item Name

The database work item name that is generated during auto-discovery is of the format *database_name*[+]:*database_type*[:*n*]. If multiple databases were backed up in a work item, then *database_name* represents one of the database names (chosen arbitrarily), and a plus sign (+) is appended to the name. *database_type* is one of oracle, informix, or sybase.

Because a database work item name must be unique, an integer was appended if necessary to make it unique.

Client Name

This is name of the client to which the work item applies, as it appears in the authorized backup list. Always use the primary complete name for the client; not an alias.

```
work item: "doc1-all", "doc1"
```

Client Name 

Filespec

The DO_FILE_LIST in this field specifies a file on the client that lists each file associated with the database that was found during database auto-discovery. This field has no character limit.

```
filespec:"DO_FILE_LIST /ext_ibm/sybase/master.dat"
```

Partition Spec

The DO_PART_LIST specifies a file on the client that lists each raw partition associated with the database that was found during database auto-discovery.

Database Type

This syntax specifies the database type:

database type: "*database_type*";

Database type is one of: informix, oracle, sybase, or none.

Database Server

This field sets the name of the database server for the work item. The syntax is:

database server: "*database_server_name*";

Database Name

This field contains one or more database names, separated by spaces.

Type

For a database work item, this field must be as follows:

type: coordinated;

Exclusion Tag

This field identifies any work item that should not be backed up concurrently with one or more other work items. For each client, the same tag value (1-63 characters) is specified for all those work items that are mutually exclusive, using this format:

```
exclusion tag: "tag";
```

For more information, see "Exclusion Tag" on page B-40.

Inclusion Tag

A single database might be backed up by two work items: one for the raw partition, and another for the files. The inclusion tag is used to coordinate back up of work items that refer to the same database. By default the inclusion tag value is the work item name. The syntax is:

```
inclusion tag: "inclusion_tag";
```

Access Time Preservation

Two alternative statements are available for clients that do not allow *invisible file access*.

(With *invisible file access*; neither atime nor ctime is updated when files are backed up. Clients that support invisible file access: Solaris 2.1 or higher for SPARC, SunOS 4.1.3 for SPARC. This setting does not apply to these clients.)

For all other clients, you can decide whether each night's backup process itself either:

- preserves the original change time (ctime), but updates each file's access time (atime) to the time of the backup, or
 - preserves the original access time (atime), but updates each file's change time (ctime) to the time of the backup
-

To preserve the original ctime, use:

```
preserve file change times during backup;
```

To preserve the original atime, use:

```
reset file access time after reading file;
```

These two options are mutually exclusive. (If both are applied, a semantic warning is issued during backup processing and then the system's default operation — preserving ctime but updating atime — is performed.)

Priority

This field assigns a priority to the work item, forcing it to run either before or after one or more other work items. See “Priority” on page B-41 for more information.

Do Not Load Balance

Load balancing schedules backups as evenly as possible across all clients. The following statement excludes the work item from load balancing:

```
do not load balance;
```

For more information, see the discussion under File System Work Item Fields, “Do Not Load Balance” on page B-43.

Backup Client Initialization Command

This is the **ebcv_shutdown** command that is run on the database client before shutdown.

The command parameters are

```
-exec_as username \  
-DBServer servername \  
-DBA database_administrator_name \  
-DBAIdent encrypted_dba_password \  
database_type
```

Do not attempt to edit the encrypted password.

Initialization Timeout

This field specifies the amount of time to wait before the shutdown and backup are considered to have failed and the attempt is terminated. The syntax is:

```
backup client initialization timeout: n day n hour n minute n second;
```

Backup Client Cleanup Command

This is the **ebcv_startup** command that is run on the database client before restarting the database.

The command parameters are:

```
-exec_as username \  
-DBServer servername \  
-DBA database_administrator_name \  
-DBAIdent encrypted_dba_password \  
database_type
```

Do not attempt to edit the encrypted password.

Cleanup Timeout

Amount of time to wait before the restart is considered to have failed and is terminated. Note that this has no effect on the actual backup. The syntax is:

```
backup client cleanup timeout: n day n hour n minute n second;
```

Buffer Sizes

The buffer sizes for the server and clients can be tuned for better backup and restore performance. The buffer sizes you choose depend on your site's configuration. Factors that influence the optimum buffer size settings are: the number and type of library units and RAM in the backup server.

The syntax of the buffer size fields is:

backup client data buffer size: *n* MB;
 backup server data buffer size: *n* MB;
 recovery client data buffer size: *n* MB;
 recovery server data buffer size: *n* MB;

Backup Start Time

This field specifies an additional qualification to the start time for nightly processing, which is specified in a crontab entry. The default crontab start of nightly processing is 11:00 p.m.

Level Map

This option overrides the backup level(s) that would normally occur for the work item, based on the template used.

For more information, see “Specify the completeness-code as any of the values below. This field applies for level 0-9 backups. The initial setting varies depending on the type of file that is being backed up (as noted in Table B-4). Generally it is a good idea to use the defaults.” on page B-45.

PC Work Item Fields

These fields are used for all PC clients: NetWare, OS2, and Windows NT Backup Clients and for OpenVMS Backup Clients. In several cases the word *netware* is embedded in the code, but the field is used for all PC clients.

Note: For details, see the appropriate client manual.

The following is a sample Windows NT work item.

```
work item: "prince:/C/", "prince"
{
  filespec: "DO_FS /C/";

  exclusion tag: "PRINCE_Harddisk0";

  use connection method: netware@3895;

  netware username: "supervisor";
```

Note: The "DO_FS" directive is not valid for NetWare and OS/2.

Work Item Name

This field defines the files to be backed up from a single client's resource. For example, the default syntax for Windows NT is:

workitem:"*servername:resource*", "*servername*"

servername is the name of the PC server. *resource* is the PC file, directory, or volume to be backed up.

Connection Method

The connection method defines the TCP/IP port on which the client is listening for a command from the backup server. The syntax is:

use connection method: netware@*nnnn*;

where *nnnn* is the port number.

NetWare default	netware@1776
OS/2 default	netware@1492
Windows NT default	netware@3895
OpenVMS default	netware@3896
NT SQL Server	netware@5600

It must be the same as the port number that is set on the PC server. If you change one, you must change them both and rerun the configuration on the PC server.

Filespec

File specification describes the files to be backed up on the client.

Netware Username

The netware username entry determines what PC user privilege can execute backups and restores. For detailed information on how this works for individual clients, see the appropriate client manual.

Netware Encrypted Password

The netware encrypted password entry provides the required password to the PC user who can execute backups and restores. Do not attempt to edit it directly. Make any changes through the EDM Backup Configuration window.

Netware Client TSA

The netware client TSA entry assigns a target service agent to the PC server.

In NetWare, OS/2, and Windows NT the format is:
servername.filesystem

Netware Client Target

The netware client target defines a PC client as a target in need of service.

Exclusion Tag

The exclusion tag ensures single-threaded processing for the client. All work items with the same exclusion tag execute sequentially. Backup processes proceed one at a time.

Backup Trailsets

A backup media set (trailset) defines all of the trails that are used for one or more work groups, (including the type of media to which the backups should be written for each level). It defines the retention period for the backed up data and related catalogs and saveset records. It also defines the level at which you want to compress the catalogs (to save disk space).

When running **ebbbackup**, you identify the schedule template to use. That template references to other types of configuration constructs:

- one or more work groups
- the trailset(s) that you want to use to back up those work groups

Each template requires one trailset (called the *primary trailset*), and can include a second (*alternate*) trailset. Generally the alternate trailset is used for off-site storage.

The backup trailset is similar to the following (for simplicity, only information for levels 0 and 9 are used):

```
backup trailset: "on site 1"
{
  use trail: "fulls-tape", dlt for level 0,
  using at most 8 clients concurrently;

  duplicate media after backup on 1 copy,

  appending to current media copy;
  use trail: "incr-dlt", dlt for level 1-9,
  using at most 8 clients concurrently;
  use trail: "baselines", EO for level B1,
  using at most 8 clients concurrently;
  use level B1 for baseline backups;
  keep backups of level 0 for: 1 year;
  keep backups of level 9 for: 3 months;

  keep duplicates of level 0 for: 1 year;

  keep duplicates of level 9 for: 3 months;
  keep backup catalogs of level 0 for: 1 year;
  keep backup catalogs of level 9 for: 3 months;
  keep saveset records of level 0 for: 1 year;
```

Note: Duplicate media information (as shown in lines 5, 6, 14, and 15 of the above example) appears only if media duplication is configured. If new media is requested for duplication, the line "using new media at each duplication" appears in place of line 6. If manual duplication is enabled, the line "manual activation of media duplication" appears after line 6.

Specify as many media sets (trailsets) as necessary to configure your site: one for each unique combination of backup levels, media, and length of time to keep the backup data, catalogs, and saveset records.

Note: The fewer backup schedule templates, trailsets, and trails you use to configure your site, the fewer backup volumes EDM Backup needs to access during processing.

Define each field as described in the sections that follow.

Backup Trailset Name

Use this field to specify a 1-63 character name for the trailset (defaults to *primary* at installation). This name must be unique within the configuration file:

```
backup trailset: "trailset name"
```

Note: The trailset name field does not end with a semicolon, but instead is followed by a brace-delimited block that defines its trails.

Changes to the trailset name take effect the next time **ebbackup** processing runs.

Use Trail

Use this field to specify the trail, including the name for the collection of media designated by the trail, the type of media that is used, the level of backups written to the trail, and the maximum number of clients that can be backed up concurrently to this trail (vs. the maximum number of clients that can be backed up concurrently for the entire server, and specified by using the global *maximum simultaneous clients* field in the server block).

You can have up to 12 trails per trailset — one for each backup level (0-9, B1, and B2). Each baseline level requires its own trail. Levels 0-9 can share trails. You generally want one or two trails for levels 0-9, plus the baseline trail(s).

This field can take two forms, and includes four fields: the trail name, media type, backup level, and number of concurrent clients. Each field is described following the syntax below:
use trail: "*trailname*", *media-type* for level *n*,
using any number of clients concurrently;

or

use trail: "*trail name*", *media-type* for level *n*,
using at most *n* clients concurrently;

Changes to the *use trail* portion of the field take effect the next time **ebbackup** processing runs. Changes to the *using ... clients concurrently* portion take effect immediately.

Trail Name

Specify the trail name as 1-11 characters. Trail names are used in volume management as part of the volume label name. For levels 0-9, the trailname and the media type are combined to form the label name. For baselines, the trail name is the entire label name.

Note: The more trails you use for each set of work groups that are backed up together, the more overhead you'll incur in terms of media management and tracking.

Media Type

Specify the type of media to which the trail is written.

Backup Level

Specify the level(s) of backup you want to write to this trail, using one of three formats:

- A single level (0 through 9, B1, or B2):
use trail: "trail_1", dlt for level 9, ...
- A range of levels:
use trail: "trail_1", dlt for levels 0-9 ...
- A compound (AND) statement that includes two levels, or ranges of levels:

```
use trail: "trail_1", dlt for levels 5 and  
levels 8-9 ...
```

Only use level B1 or B2 if you purchased HSM, to indicate whether you want to maintain a level B1 or B2 backup. If you want to maintain two baseline levels, specify one trail as B1 and the other as B2.

Note: Never write level B1 and B2 backups to the same trail.

If you identify more than one trail for a particular level, the last specification overrides the others. In the following example, EDM Backup uses the *incr-dlt* trail for level 9 and *mid-dlt* for level 5, even though levels 9 and 5 were included in the range of levels specified for the (first) *fulls-tape* trail:

```
use trail: "fulls-tape", dlt for levels 0-9,  
using at most 8 clients concurrently;  
use trail: "incr-dlt", dlt for level 9,  
using at most 8 clients concurrently;  
use trail: "mid-dlt", dlt for level 5,  
using any number of clients concurrently;
```

Make sure to include every level of backup you want to run in at least one trail. For example, if you want to perform level-5 backups from the command line, you must define level 5 somewhere in the trailset. It is best always to include a range of levels (for example, 0 through 9) for each trail, even though some levels may never be used. (Automatically scheduled backups only use levels 0 and 9, for example.)

Maximum Number of Concurrent Clients

Specify the maximum number of work items (not clients) that can be backed up at the same time to this trail, in the range 1-24 (defaults to 8 at installation):

```
using at most n clients concurrently;
```

Alternatively, specify as many work items as possible (up to the number indicated by the server block's *maximum simultaneous clients* field):

```
using any number of clients concurrently;
```

The recommended setting is from 2 to 8:

- Use a lower number (for example, 2-3) if the trail is used mostly for full backups, or for incrementals that include a lot of file data (vs. extended inodes only)
- Use a higher number (up to 8) if the trail is used mostly for sparse incremental backups
- Choose a value in between if the trail is used for some combination of full backups and sparse incremental backups

This setting is used in conjunction with the system-wide (server block) *maximum simultaneous clients* field, to control the use of system resources. Regardless of the trail-level setting, there will never be more work items processed concurrently than are specified by the system-wide value.

For example, if you are using two trails that each specify four work items but your system-wide work-item limit is six, there can never be more than six work items writing to the two trails combined, at any one time. In this situation, EDM Backup:

- Starts backups on four work items using the first trail.
- Starts backups on two more work items using the second trail, bringing the total number of work items processed by the server to the limit.
- Starts a new work item each time an active work item finishes, keeping the total number of work items to six. In juggling the work items, EDM Backup does its best to allot equal resources to all scheduled work items.

This example is illustrated in Figure B-2 (where, for simplicity, assume one work item per client).

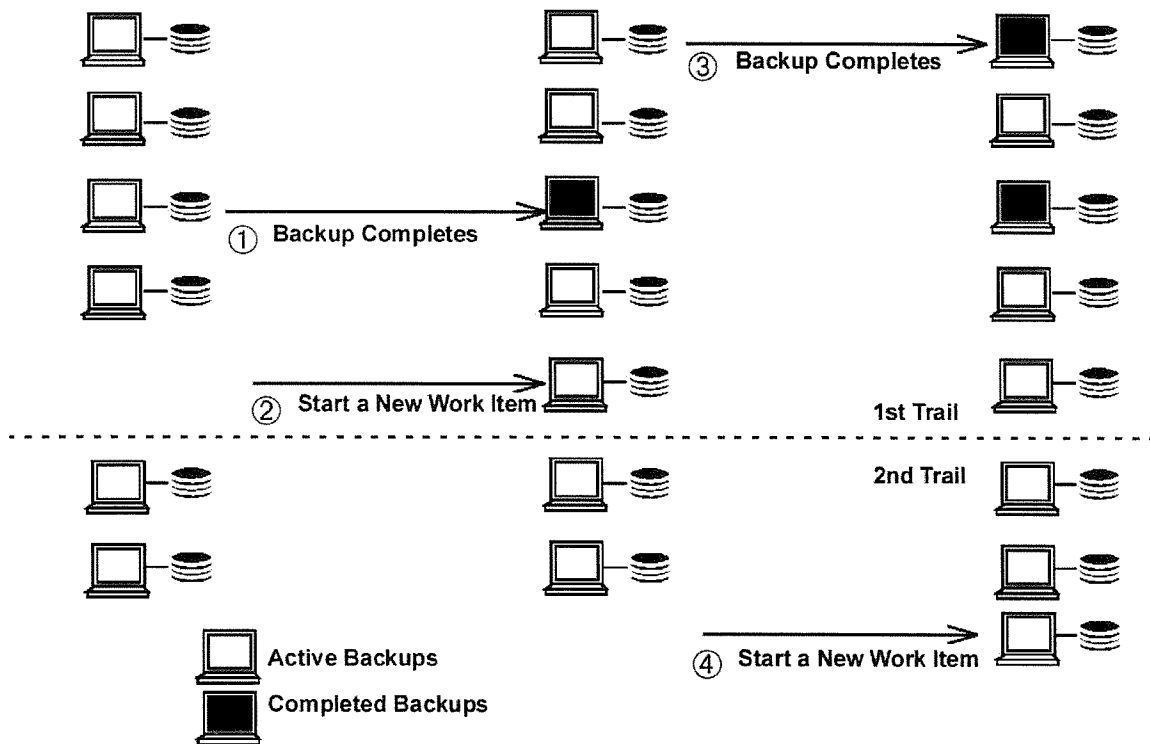
Figure B-2

Backup Processing

EDM Backup starts six work-items — 4 use the 1st trail, 2 use the 2nd.

When a work item finishes (1), EDM Backup starts a new work item (2).

As each work item finishes (3), EDM Backup starts a new one (4).



Use Level B1 / B2 For Baseline Backups

If you are using automatic scheduling, the scheduler writes a baseline backup to a B1 level trail. If an alternate trailset is configured with a B2 level trail, the autoscheduler writes an alternate night baseline backup to a B2 level trail.

Note: The entry "use level B1 for baseline backups;" in the eb.cfg file is ignored by the backup scheduler. The B2 level is always used for baselines on alternate night as long as the alternate trailset is configured with a B2 baseline trail.

Specify at most one baseline level per trailset. If you include this field, make sure to specify a *use trail* statement for the baseline backup.

Omit this field if you are not writing baseline backups to the trailset, or if you plan to request all baseline backups via custom or command-line scheduling. This field is ignored with custom and command-line scheduling.

Changes to this field take effect the next time EDM Backup schedules work items.

Keep Backups

Use this field to indicate how long to retain the backup data before reusing the backup media. Specify this field by using the following format, repeating it as many times as necessary to configure expirations for your site:

Note: If you identify more than one trail for a particular level, the last specification overrides the others.

keep backups of level *n* for: *time-period*;

where:

- *n* specifies the backup level (s) to which the setting applies (0-9), by using one of three formats as described for the *use trail* statement (a single level, range of levels, or compound statement that combines levels).

Generally you define the same levels (or set of levels) here as are specified via *use trail* statements for the same trailset.

Note: This specification does not apply for baseline backups.

- *time-period* defines the amount of time to retain the backups for this level, specified as an integer followed by an appropriate *units* value.

Units

second(s)

day(s)

week(s)

month(s)

year(s)

forever (*do not include an integer*)

You can expire backups immediately:

```
keep backups of level 2 for: 1 second;
```

Or keep them indefinitely:

```
keep backups of level 0 for: forever;
```

You can combine the time units, as in the following example:

```
keep backups of level 9 for: 1 year; 6 months;
```

Always expire backup data before (or at the same time as) the corresponding saveset records, but after (or at the same time as) the corresponding catalogs. An exception occurs if you specify *keep backups forever*, in which case you can expire the saveset record without expiring the media. Make sure to cover every backup level that is used.

The recommended setting (and the initial value at installation) is 1 year for level-0 backups, and 3 months for levels 1-9:

```
keep backups of levels 0-9 for: 3 months;
```

```
keep backups of level 0 for: 1 year;
```

Note: If you identify more than one expiration for a particular level, the last specification overrides the others.

If you omit this field, it defaults to *forever*.

Changes to this field take effect immediately, but do not affect any currently running work items.

Keep Duplicates

Use this field to indicate how long to retain a duplicate of backup data before reusing the duplicate's media. Specify this field by using the following format, repeating it as many times as necessary to configure expirations for your site:

Note: If you identify more than one trail for a particular level, the last specification overrides the others.

```
keep duplicates of level n for: time-period;
```

where:

- *n* specifies the backup level(s) of the duplicate, to which the setting applies (0-9).
- *time-period* defines the amount of time to retain the duplicates for this level, specified as an integer followed by an appropriate *units* value.

Note: Refer to “Keep Backups” on page B-65 for more information about these fields.

You can expire duplicates immediately:

```
keep duplicates of level 2 for: 1 second;
```

Or keep them indefinitely:

```
keep duplicates of level 0 for: forever;
```

You can combine the time units, as in the following example:

```
keep duplicates of level 9 for: 1 year 6 months;
```

Note: You must expire duplicate data before (or at the same time as) the corresponding backup data. Refer to “Rejecting a Mount Request” on page 9-29 for more information.

Keep Backup Catalogs

Use this field to indicate how long to keep the backup catalogs created for each level 0-9 backup. Always expire the catalogs before, or at the same time as, the media (above).

The format of this field is:

```
keep backup catalogs of level n for: time-period;
```

Where *n* and *time-period* are specified as described above for the keep backups setting. Make sure to cover every backup level used.

The recommended setting (and the initial value at installation) is 1 year for level-0 backups, and 3 months for levels 1-9:

```
keep backup catalogs of levels 0-9 for: 3 months;  
keep backup catalogs of level 0 for: 1 year;
```

Note: If you identify more than one expiration for a particular level, the last specification overrides the others.

You should expire the catalogs for incremental backups aggressively to free up disk space, because:

1. You rarely restore files from older incremental backups and
2. You can reconstruct the catalogs from the backups, if necessary

If you omit this field, it defaults to *forever*.

Note: See Chapter 10 "Magnetic Disk Concepts".

Changes to this field take effect immediately, but don't affect any currently running work items.

Keep Saveset Records

Use this field to indicate how long to keep the saveset records created for work items written to this trailset. Saveset records are necessary during a restore, to locate backup data and catalogs. Because of this, you can't generally expire saveset

records before either the backup data or the catalogs. An exception occurs if you specify *keep backups forever*, in which case you can expire the saveset record.

The format of this field is:

```
keep saveset records of level n for: time-period;
```

Where *n* and *time-period* are specified as described above for the keep backups setting, except that you can specify B1 and B2 as well as levels 0-9. Make sure to cover every backup level used.

The recommended setting (and the initial value at installation) is 1 year for level-0 and baseline backups, and 3 months for levels 1-9:

```
keep saveset records of levels 0-9 for: 3 months;  
keep saveset records of level 0 for: 1 year;  
keep saveset records of level B1 for: 1 year;
```

Note: If you identify more than one expiration for a particular level, the last specification overrides the others.

If you omit this field, it defaults to *forever*.

Changes to this field take effect immediately, but don't affect any currently running work items.

Backup Catalog Delta Level

Use this field to specify the backup level at which you want EDM Backup to consolidate catalogs (1-9). The recommended setting (and the initial value at installation) is 9:

```
backup catalog delta level: 9
```

If you omit this statement, the backup catalogs are consolidated at level 1.

When EDM Backup consolidates the catalogs, it turns all the catalogs for the level specified — as well as any numerically higher levels — into deltas, which only contain information that

differs from the more recent catalogs. If you specify a delta level of 5, EDM Backup compresses the catalogs for level 5-9 backups.

EDM Backup recreates the full catalog from a delta as necessary during a restore, by adding back the information that was originally compressed out. It uses as input all subsequent catalogs for the work item, up to (and including) the most recent catalog.

The latest catalog for any particular work item is always uncompressed, to provide fast access during restore processing. For the same reason, level-0 catalogs are always uncompressed.

Changes to this field take effect immediately, and could affect work items currently being backed up, but not any whose catalog is currently being processed. Since this field is used by catalog processing after backup processing, it should take effect on the next catalog to start being processed by **ebcatalogd** and **ebcatd**. This could be the currently running work items, since their catalogs have not been processed yet.

Backup Template Fields

A backup schedule template describes how to back up a group of clients, grouping together work group(s) and trailsets. You can have as many backup schedule templates as you want, but you must have at least one. The template specifies:

- The list of work group(s) to back up.
- The rotation period to use and date to start the first rotation.
- The trailset(s) on which to store the backup data for clients in the work group(s). Each backup template has a *primary* trailset, and may have an optional *alternate* trailset. Generally the alternate trailset is used for off-site storage.
- Whether to force all baseline backups to finish before any level 0-9 backups can start (available only if you've purchased HSM).

- Whether to recreate the baseline backup if necessary (available only if you've purchased HSM).
- Information about the log files and completion reports.
- Scheduling specifications.

Specify the template fields as described in the sections that follow.

Backup Template Name

Specify this field as the 1-63 character name for the template:
`backup template: "template name"`

Note: The template name field does not end with a semicolon, but is followed by a brace-delimited block that defines its specifications.

The *template name* must be unique within the configuration file. You'll reference the template name when you invoke **ebbackup** to start a backup.

```
backup template: "sales-all"
{
  work group list: "sales", "local";
  begin trailset rotations on: Dec 13, 1993;
  rotation period: 7 days;
  primary trailset: "on site 1", use new volume on
  each rotation;
  alternate trailset: "off site 2", use new volume on
  each rotation;
  logging level: stats;
  server logfile: "sales-all_template.log", 256k;
  backup completion script: "mailok";
  backup failure script: "mailerr";
  do all baseline backups before normal backups;
  recreate baseline if needed;
  schedule:
  {
    /* standard rotations; */
    /* full during weekends rotations; */
    weekday backup shift is 8 hours;
    weekend backup shift is 24 hours;
    level 9 on Monday, Tuesday, Wednesday, Thursday,
    Friday;
    for "sales", level 0 on Saturday;
    for "local", level 0 on Sunday;
  }
}
```

Note: If you remove a template from the configuration file, it remains in the schedule, as do any work items scheduled for that template. To remove a template completely, first remove it from the configuration file, then use **ebbackup** with the **-retire** option to delete it from the schedule.

Changes to the template name take effect the next time **ebbackup** processing runs.

Work Group List

Use the *work group list* to specify the work groups that are backed up together using this template. This is a shorthand method of identifying each work item individually. See “Work Group Fields” on page B-31 for information about defining work groups.

The format of this field is:

```
work group list: "work group", "work group",
..."work group";
```

Changes to this field take effect the next time **ebbackup** processing runs.

Begin Trailset Rotations

This field specifies when EDM Backup should begin using the template (for a new template, it defaults to the date the template is added). It applies for all work items in the template, and for both automatic and custom scheduling (but not for command-line scheduling).

If you want to create a new template, but do not want to start using it yet, use this field with the date when the template is to be used. The format of this field is:

```
begin trailset rotations on: date;
```

Where *date* takes one of three formats:

Format	Example
dd-mmm-yy	13-dec-99
mm/dd/yy	12/13/99
mmm dd, yy	Dec 13, 99
mmm dd, yyyy	Dec 13, 1999

Leading zeros are not required. EDM Backup interprets the century using standard UNIX conventions:

- Years in the range 70-99 are in the 20th century (i.e., 19yy)
- Years in the range 0-37 are in the 21st century (i.e., 20yy)
- Years in the range 38-69 are illegal — EDM Backup can't create timestamps for these years because of a UNIX restriction

Changes to this field take effect the next time EDM Backup schedules work items.

Rotation Period

This field indicates how often each client should receive a level 0 backup (autoconfigured to every 14 days). For automatic scheduling, it's used by EDM Backup to compute the schedule of level 0 and level 9 backups, ensuring at least one level 0 backup during each rotation. For custom scheduling, it serves as the point of reference when defining the backup schedule (for example, run level 0 backups on the 1st day of each rotation, and run level 9 backups on days 4, 8, and 12).

The rotation period also indicates when EDM Backup should start a new media rotation for the template. By default, EDM Backup starts writing to a new set of media on day 1 of each new rotation. You can override this either in the statement that defines what trailsets to use (described under "Primary and Alternate Trailsets" on page B-75), or on the command line, when you run **ebbackup**.

The format of this field is shown below:

rotation period: *n time-units*;

Where:

- *n* is any integer greater than 1. The most common rotation periods are 7 days, 14 days, or 28 days. For automatic scheduling:
 - Use a 7-day rotation if restore time is important and you can afford the resources necessary to maintain a full backup every seven days.

- Use a 14-day rotation (the installation default) unless you have a good reason to use a different setting. The 14-day rotation conserves media and generally offers a good restore rate.
- Use a 28-day rotation to minimize the use of backup media, or for filesystems that change little over time (such as / and /usr). Restoring files will be slowest with this schedule, because there may be more incremental backups to read; however, the effect should be minimal if your data changes little over time.
- *time-units* defines the unit of time in which the rotation period is specified:

Units

day(s)

week(s)

month(s)

year(s)

For example, to automatically perform a full backup every 7 days on the clients named by the work group, enter:

```
rotation period: 7 days;
```

Changes to this field take effect the next time EDM Backup schedules work items.

Primary and Alternate Trailsets

A template must specify one trailset, and can specify two, known respectively as the *primary* and *alternate* trailsets.

Specify the trailset(s) using this format:

```
primary trailset: "trailset name",  
use new volume on each rotation;  
alternate trailset: "trailset name",  
use new volume on each rotation;
```

Where:

- The *trailset name* identifies the trailset to which backups are written, and must specify a name that's defined already in the configuration file (set to "primary" at installation).
- The clause *use new volume on each rotation* directs EDM Backup to start a new volume(s) at the beginning of each rotation period. This separates backups from different rotation periods into distinct volume sets.

If you specify an alternate trailset, EDM Backup switches between the two trailsets each day, scheduling the primary trailset on odd-numbered days with respect to the template's rotation period, and the alternate trailset on even-numbered days. With automatic scheduling, this provides a second separate set of backups on alternating nights. If the rotation period specifies an even number of days, the two sets of backups are nearly the same, differing only by one day's worth of data.

With custom scheduling it can do the same, but it's up to you to schedule identical backups on alternating days. If you specify an alternate trailset with custom scheduling, make sure to define two complete (and exact) schedules, running the same level on each of two consecutive days. By scheduling the following levels to run on the days shown, you'll create two identical trailsets that differ by one day's worth of data.

Day	Level	Trailset Used
Monday	0	Primary
Tuesday	0	Alternate
Wednesday	9	Primary
Thursday	9	Alternate
Saturday	9	Primary
Sunday	9	Alternate

You can spread out the backups more, as long as you specify one of the two identical backups on an even day with respect to the rotation cycle, and one on an odd day.

Changes to this field take effect the next time **ebbackup** processing runs.

Logging Level

Use this field to specify the level of logging messages written to the file specified via the server log file field when this template is backed up.

Specify the logging level using this format:

`logging level: logging level;`

Where *logging level* is specified as follows (defaults to *stats* at installation).

Code	Description of Messages Logged
none	No messages.
errors	Logs errors, including device and communication errors.
stats	Logs statistics as well as the information for the <i>errors</i> level. The statistics include the amount of data saved, the time it took to save the data, when the backup started and finished, and so on.
debug	Logs debugging information as well as the information for the <i>stats</i> level. Only use this level at the direction of customer support.
per file	Logs several lines, including <i>debug</i> -level information, for each file that's backed up. Only use this level at the direction of customer support. It slows backup throughput and can result in a huge log file.

Changes to this field take effect immediately, but don't affect any currently running work items.

Server Log File

Use this field to specify the name and maximum size of the template's log file. This file is stored on the server as `/usr/epoch/EB/log/file-name` and records information as directed by the *logging level* field. It provides an audit trail of all backup activities for the template.

The format is:

```
server logfile: "log file name", file-size;
```

Where:

- *log file name* identifies the log file on the server (stored in the `/usr/epoch/EB/log` directory, set initially to `default_template.log`, where "default" is the template name). You can create a different log file for each template or you can share a single log file across all templates. If you specify a separate log file for each template, use a name that identifies the template.
- *file-size* indicates how large the file can get before the oldest data is expired. When the file reaches the specified size, EDM Backup locates the oldest data in the file and expires 10 percent of that data to free up space for new information. Specify this field as a number of bytes followed by a unit code (e.g., 500KB, set to 256K at installation). Specify the unit code as follows:

Unit Code	Measure
k, K, kb, or KB	kilobytes
m, M, mb, or MB	megabytes
g, G, gb, or GB	gigabytes

To allow the file to grow until it's as large as permitted by physical storage space, specify *no limit* (and no other options). By not setting a limit on the file, it becomes a permanent audit trail of backup operations for the template:

```
server logfile: "default_template.log", no limit;
```

Note: If you don't limit the file size, it may become too large to manage. Also, in HSM systems it won't be staged.

This example logs messages for the *sales-all* template, and has a maximum file size of 256KB:

```
server logfile: "sales-all_template.log", 256KB;
```

If you omit this setting, no log file is maintained for the template.

Changes to this field take effect immediately, but don't affect any currently-running work items.

Backup Completion Script

EDM Backup generates completion reports describing each successful backup. There is one completion report per template. EDM Backup forwards these reports to the script file indicated using this parameter. The script file, in turn, dispatches the reports according to how the script is defined.

As installed, EDM Backup directs completion reports to the supplied "mailok" script on the server (/usr/epoch/EB/config/mailok). Depending on how that script is defined, it sends the reports to specific administrators (generally those defined in the list of backup administrator usernames), to the log file, or to both.

You can change the script by entering a new script name in the **Advanced Options** popup window in the **Schedule** tab of the EDM Backup Configuration window.

If you specify a relative pathname, the file is assumed to be in /usr/epoch/EB/config. If you specify an absolute pathname, the file can be located anywhere.

For a sample report, see "Backup Completion Reports" on page 16-29.

If you this setting is blank, no completion reports are generated.

Changes to this field take effect immediately.

Backup Failure Script

EDM Backup creates backup failure reports for each work item whose backup fails. It forwards these reports to the script file indicated using this parameter. The script file, in turn, dispatches the reports according to how the script is defined.

By default, EDM Backup directs failure reports to the EDM-supplied “mailerr” script on the server (/usr/epoch/EB/config/mailerr). Depending on how that script is defined, it mails the reports to specific administrators (generally those defined in the list of backup administrator usernames), to the log file, or to both.

You can change the script by entering a new script name in the **Advanced Options** popup window in the **Schedule** tab of the EDM Backup Configuration window.

If you specify a relative pathname, the file is assumed to be in /usr/epoch/EB/config. If you specify an absolute pathname, the file can be located anywhere.

For a sample report, see “Backup Failure Reports” on page 16-31.

If this setting is blank, no failure reports are generated.

Note: You can leave this field blank to reduce the amount of EDM Backup mail. The backup completion script (*mailok*) reports include all types of activity, including failures. You can refer there for information about failures.

Changes to this field take effect immediately.

Do All Baseline Backups Before Normal Backups

Use this statement if you’ve purchased HSM, and want to ensure that *all* baseline backups finish for the template before the level 0-9 backups start for *any* work items backed up by the template:

```
do all baseline backups before normal backups;
```

This feature is useful when baseline backups consume most of the server's resources, which can happen if significant amounts of magnetic disk space are required to hold temporary files. It can also alleviate disk thrashing. On the downside, it can slow overall throughput.

With this statement omitted (the default at installation), EDM Backup finishes the baseline backup *for each work item* before starting the level 0-9 backup for that same work item.

Note: This specification applies for custom and automatic scheduling, but is not used for command-line scheduling.

Changes to this field take effect the next time **ebbackup** processing runs.

Recreate Baseline if Needed

Include this statement if you've purchased HSM, and you want to automatically re-baseline any file for which the existing baseline copy was substituted for the staged copy during a restore:

```
recreate baseline if needed;
```

If you don't want to re-baseline files automatically, use the following statement instead (or don't include either statement). This is the default setting at installation:

```
do not recreate baseline if needed;
```

If you choose to re-baseline files automatically, **ebbackup** checks each file when it runs a baseline backup, to see if the staging id in the file's extended inode is the same as the baseline id. These two ids are only the same if the baseline copy of the file was substituted for a missing (or damaged) staged version. If the ids are the same, **ebbackup** automatically generates a new baseline copy of the file, then updates the baseline id so that it points to the new copy.

Changes to this field take effect the next time **ebbackup** processing runs.

Schedule

The schedule tells EDM Backup when to run backups for the template, and how much time to commit to backup processing each day. For custom scheduling, it defines when to run each backup level (and optionally, when to run each level *for a specific work group*).

Some scheduling options apply for automatic scheduling and some for custom scheduling, as follows:

Table B-6

Scheduling Fields

Option	Applicable for:	
	Automatic	Custom
Standard rotations	yes	no
Full during weekends rotations	yes	no
Weekday backup shift	yes	no
Weekend backup shift	yes	no
[for <i>work-group-name</i>] level <i>n</i> on days ...	no	yes

The schedule block looks like this:

```
schedule:
{
/* standard rotations; */
/* full during weekends rotations; */
weekday backup shift is 8 hours;
weekend backup shift is 24 hours;
level 9 on Monday, Tuesday, Wednesday, Thursday,
Friday;
for "sales", level 0 on Saturday;
```



```
for "local", level 0 on Sunday;
}
```

Note: A level map takes precedence over any scheduling defined using this block. For example, if you'd normally create one level 0 backup and the rest level 9 backups during the rotation (as for automatic scheduling), but a work item specifies a level map of "BB0000000000", each level 0-9 backup for that work item will run as a level 0 backup.

Specify each field as described below. In general, try to schedule backups when system use is low.

Standard vs. Full-During-Weekends Rotations

Most sites let EDM Backup schedule backups automatically, based on the settings in the configuration file. If this is the case at your site, choose one of these two scheduling directives to:

- Turn on automatic scheduling (selecting either directive does this).
- Specify how to perform backups. Indicate the statement you want by placing comment markers (`/* ... */`) around the other statement:

```
standard rotations;
/* full during weekends rotations; */
```

Specification	Description
Standard rotations	Schedules backups so that some portion of the clients receive a full backup each day
Full during weekends rotations	Schedules backups so that all clients that require a full backup receive that backup on Saturday or Sunday, when possible, with any incrementals running on weekdays

Changes to this field take effect the next time EDM Backup schedules work items.

Specifying Backup Shifts

If you're using automatic scheduling, use these two fields to indicate how much time you want to allow EDM Backup to run each day. This serves as a guideline, or goal, when EDM Backup schedules the work items for backup. Specify the time for each weekday (Monday-Friday), then for each weekend day (Saturday and Sunday):

```
weekday backup shift is hh hours mm minutes;
weekend backup shift is hh hours mm minutes;
```

Specify *hh* as a number of hours (an integer in the range 1-24), and *mm* as a number of minutes (1-60; include only if necessary):

```
weekday backup shift is 8 hours 30 minutes;
weekend backup shift is 24 hours;
```

Note: The backup shift specifications are not binding, but serve only as a target for the amount of time committed to backup processing each day.

Changes to this field take effect the next time **edbackup** processing runs.

Scheduling Custom Backups (Level *n* on Days...)

If you're scheduling custom backups, use this statement to:

- Turn on custom scheduling (happens automatically when you include one or more custom-scheduling statements and you omit both statements used to turn on automatic scheduling)
- Define when you want to run the backups, repeating this statement as necessary to configure your site:

```
[ for "work-group-name" ] level n on [ day(s) ]
days;
```

You can specify each work group individually, or you can combine the work groups for the template into a single statement. You might want to combine the work groups for one or more level(s), but separate them for another level(s):

```
level 9 on Monday, Tuesday, Wednesday, Thursday,
Friday;
```

```
for "sales", level 0 on Saturday;  
for "local", level 0 on Sunday;
```

Specify each syntax field as follows:

- The *for work-group-name* clause applies if the statement is for a specific work group, and specifies the name of the work group exactly as it appears in the template's work-group list (defaults to all work groups in that list if you omit this clause). If two statements apply for the same work group, EDM Backup uses the most restrictive specification (the one that's specific to the work group). You might use one statement to schedule all work groups, then add statements for each exception to the "rule":

```
level 9 on days 1-14;  
level 0 on days 7, 14;  
for "sales", level 0 on day 1;
```

- The *level n* clause identifies the backup level you want to run on the day(s) specified (0-9, B1, or B2). If more than one level is specified for the same day, EDM Backup performs the lowest level, numerically (e.g., level 0 instead of level 5).

If the template specifies an alternate trailset, make sure to schedule two level-0 backups for each work group — one on an even-numbered day and one on an odd-numbered day, with respect to the rotation cycle. (Remember that the two trailsets are backed up on alternating days.)

Note: Specify baseline backups explicitly. EDM Backup does not run baselines automatically when you use custom scheduling.

- *days* indicates the days on which you want to run backups for this level (and optionally, for this work group).

If you don't request backups for a particular day, EDM Backup won't perform backups on that day (for this template).

Note: If you schedule custom backups, don't specify either of the directives for automatic scheduling (described under "Specifying Backup Shifts" on page B-84). This causes an error.

Specify the days as follows:

Days Specification	Description
<i>a single integer</i>	To represent that day in the rotation period (1 for the first day, 2 for the second day, and so forth up to the number of days specified in the rotation period for the template).
a range of integers	To represent a range of days in the rotation period <i>only for a rotation period this is NOT a multiple of 7 days long</i> (e.g., 1-6 for days 1, 2, 3, 4, 5, and 6).
name of day(s)	To represent a specific day of the week (<i>that is, for a rotation period of 7 days or a multiple of 7 days long</i>). Spell out the full name of the day (Saturday, Sunday, etc.). You <i>cannot</i> specify a range of names — each day must be listed separately. You can describe a specific occurrence of a day within the rotation period, by prefacing the day with a number (2nd Saturday, 3rd Monday, etc.).

Changes to this field take effect the next time EDM Backup schedules work items.

Startup Parameters

Use the startup parameters block to specify when to perform the first (level 0) backup for a new client. It's important to perform the first level 0 backup as soon as possible, because no incrementals run until the first level-0 backup is performed.

This block applies for automatic scheduling only, and is ignored with custom scheduling. It looks like this:

```
startup parameters:
{
perform initial full backup on scheduled day;
```

```
/* perform initial full backup as soon as possible;  
*/  
}
```

Specify one of the following in the EDM Backup Configuration window:

- Specify *perform initial full backup on scheduled day* to perform the initial full backup on some portion of the newly-installed clients each day during the first rotation period. This setting distributes the work load as evenly as possible across the rotation period, and is the recommended (and initially configured) approach. By the end of the first rotation, every client will have a level-0 backup.
- Specify *perform initial full backup as soon as possible* to run the initial full backup on all newly-installed clients during the first backup run after those clients are installed. When you use this option, EDM Backup performs initial full backups on every new client, one after the other. The impact on the system is dictated by the total amount of client data that needs to be backed up.

You may want to specify *perform initial full backup on scheduled day* when you first configure your system, then change it to *perform initial full backup as soon as possible* after the initial backups are taken (for use as new clients are added).

With either method, the client gets backed up according to its normal rotation schedule after the initial level-0 backup is completed.

Note: This specification only applies for the first backup that's run after a client is installed. It has no effect on subsequent processing.

Changes to the startup parameters block take effect the next time backup scheduling occurs.

C Volume Management Configuration Files

During system startup, the Volume Manager starts each Library Manager that is specified in the VM configuration file. A Library Manager is added to the VM configuration file when you configure a Library Manager by using the **lmconfig** utility.

This chapter provides detailed information about the Volume Manager and Library Manager configuration files.

The chapter contains the following sections:

- Volume Manager Configuration File
- Library Manager Configuration Files

See Chapter 17 “Configuring Library Managers” for information about using the **lmconfig** utility.

Volume Manager Configuration File

When you install the software, a default configuration file (vm.cfg) is added to the /usr/epoch/etc/vm directory.

EMC sets the recommended values for parameters in the vm.cfg file and, in most cases, these do not require modification. The only time vm.cfg needs to be modified is when you add a Library Manager; lmconfig manages this when you run the script to configure a Library Manager.

Figure C-1 provides a sample vm.cfg file. (Note that comment lines begin with the # character.)

Table C-1 describes the parameters and default values in the vm.cfg file.

Figure C-1

Volume Manager Configuration File

```
# The following is a default configuration file for a vmdae
#
# Limitations:
# o LM_HOST must always be localhost
# o VM_ALLOW_DUP* must always be no
# o VM_METRIC* are not used yet
# o VM_LOCAL_ONLY_ACCESS must always say no
# o VM_AUTH_TYPE may only by RPC_UNIX
#

VM_NAME : vm
VM_ROOT : /usr/epoch/etc/vm

VM_SYSLOGLEVEL : concise

VM_WATCHDOG_LMS : yes
VM_CATALOG_DIR : /usr/epoch/etc/vm
```



```
#

VM_ALLOW_DUP_BARCODE_IMPORT : no
VM_ALLOW_DUP_SEQ_IMPORT : no
VM_METRIC_INTERVAL : 60
VM_METRIC_EXPIRATION : 33
VM_LOCAL_ONLY_ACCESS : no
VM_AUTH_TYPE : RPC_UNIX

# Field for the offline mount user routine for new media request
VM_NEW_MEDIA_USR_ROUTINE : /bin/true

VM_VOLFILE_DIR : /usr/epoch/etc/vm

# The CLOG_SIZE value determines how large the vm's debugging
# circular clog size is allowed to be. The circular log or clog
# file is only written to when the vmdaemon is in debugging
# mode. A CLOG_SIZE OF 0 MEANS INFINITE, I.E. NOT CIRCULAR
VM_CLOG_SIZE : 10485760

# The VM_ERASE_LIMIT controls if and how the vmdaemon limits
# the number of concurrent volume erases per library unit.
# NONE : no limit on the number of concurrent volume
# erases in an LU
# NUM_DRIVES : concurrent erase daemons are limited to the number
# of drives in the LU containing the volumes.
# HALF_OF_DRIVES : concurrent erase daemons are limited to half of
# the number of drives in the LU containing the volumes.
# If the number is odd, it will be rounded up.
VM_ERASE_LIMIT : NUM_DRIVES

#The VM_FIND_AVAIL_MEDIA_INTERVAL determines after how many
# VM activities search for available media will be done.
VM_FIND_AVAIL_MEDIA_INTERVAL : 25

VM_DUP_NUM_ACTIVE : 1
VM_DUP_STATE : enabled
VM_DUP_TAPE_PAD : 10
```

```
#BEGIN_offline_0
LM_START : 0
LM_NAME: offline_0
LM_HOST : edmdoc
LM_EXEC_OPT : ""
LM_EJECT_DEST : offsite_0
LM_END : 0
#END_offline_0
#BEGIN_offsite_0
LM_START : 0
LM_NAME: offsite_0
LM_HOST : edmdoc
LM_EXEC_OPT : ""
LM_EJECT_DEST : offline_0
LM_END : 0
#END_offsite_0
#BEGIN_qntm_x700_0
LM_START : 0
LM_NAME: qntm_x700_0
LM_HOST : edmdoc
LM_EXEC_OPT : ""
LM_EJECT_DEST : offline_0
LM_END : 0
#END_qntm_x700_0
#BEGIN_hp_c17xx_0
LM_START : 0
LM_NAME: hp_c17xx_0
LM_HOST : edmdoc
LM_EXEC_OPT : ""
LM_EJECT_DEST : offline_0
LM_END : 0
#END_hp_c17xx_0
```

Table C-1

Volume Management Configuration Parameters

Parameter	Description
VM_NAME	Identifies the name of the Volume Manager. The default is vm. This name is used in system log messages to identify the process that generated the message.
VM_ROOT	Identifies the full pathname of the vmdaemon. The default is set to /usr/epoch/etc/vm.
VM_SYSLOGLEVEL	Sets the logging level. The default is concise, which logs only critical messages. Note: This setting is overridden at system startup.
VM_WATCHDOG_LMS	Enables the vmdaemon to monitor all Library Manager daemon (lmd) processes and to restart any that fail. The default is yes.
VM_CATALOG_DIR	Specifies the location of the volume and template catalogs. The default is /usr/epoch/etc/vm.
VM_ALLOW_DUP_BARCODE_IMPORT	Determines whether duplicate barcodes are allowed within a server. The default is no, which does not allow you to import a volume if the same barcode ID is already in the volume catalog. This ensures that all barcoded volumes remain unique within a server.
VM_ALLOW_DUP_SEQ_IMPORT	Determines whether duplicate volume sequence numbers are allowed within a server. The default is no, which does not allow you to import a volume if the same volume sequence number is already in the volume catalog. This ensures that all volume sequence numbers remain unique within a server.
VM_METRIC_INTERVAL	Specifies the frequency, in seconds, that the vmdaemon records metrics. Metrics enable you to gather performance results for use in trend analysis. For example, the vmdaemon could track mount faults once per hour and the data enables you to determine the peak of certain activities. The default is 60; the value is ignored, which indicates that no metrics are recorded.
VM_METRIC_EXPIRATION	Specifies the time period, in days, when metric data should be expired. The default is 33 days; this parameter is ignored.

Table C-1

Volume Management Configuration Parameters (Continued)

Parameter	Description
VM_LOCAL_ONLY_ACCESS	Specifies whether the vmdaemon (as an RPC server) can receive service requests from RPC clients on the network. Examples of RPC clients include: the EDM Library Unit Manager GUI, EDM Backup and HSM software, and volume management's CLI. The default is No.
VM_AUTH_TYPE	Sets the server authentication policy to RPC_UNIX, which makes the vmdaemon check user and group permissions for the session. This allows the vmdaemon to restrict functions to clients. A value of RPC_NONE disables checking of permissions.
VM_NEW_MEDIA_USR_ROUTINE	Provides a place holder for the user script that is executed in response to receiving an offline media request for available media. This user-specified routine is executed once for each media request. For example, a user can include sending email upon receiving the request in this routine.
VM_VOLFILE_DIR	For EMC internal use only. Specifies the pathname that contains pseudo media types. The default is /usr/epoch/etc/vm. Pseudo media types are created by using files on a traditional filesystem.
VM_CLOG_SIZE	<p>Sets the maximum size of the vmdaemon debugging circular log (clog) file. The default is 10485760 (10 MB). The vmdaemon writes to this file while in debug mode.</p> <p>Note: Debug mode is enabled by default at system startup.</p> <p>Note: EMC Customer Service requires VM (and LM) clog files to debug a problem effectively. It may be necessary to increase the size of the clog file to prevent the vmdaemon from overwriting the contents of this file.</p> <p>Note: Setting the default to zero is not recommended; the circular log file will not have any size limitation, which can cause serious disk space issues.</p> <p>To run in debug mode, start the vmdaemon with the -d option or by sending a special RPC request to the vmdaemon. Once in debug mode, the vmdaemon writes to the clog file and continues to wrap until it reaches the maximum specified size. When the file reaches the maximum specified size, the vmdaemon writes to the top of the file.</p>

Table C-1

Volume Management Configuration Parameters (Continued)

Parameter	Description
VM_ERASE_LIMIT	<p>Specifies how many drives are allowed to perform volume erasures per optical library unit.</p> <p>Note: This parameter is ignored for tape media.</p> <p>Values include:</p> <p>NONE no limit to the number of concurrent volume erasures in a library unit.</p> <p>HALF_OF_DRIVES limits the number of concurrent volume erasures to half of the drives in the library unit.</p> <p>NUM_DRIVES (the default) limits the number of concurrent volume erasures to the number of drives in the library unit containing volumes.</p>
VM_FIND_AVAIL_MEDIA_INTERVAL	<p>Indicates that if a queued request exists, a search for available volumes from the catalog is made only after every <i>n</i> number of VM activities. (However, a search is done immediately after particular activities such as import, inject, or inventory.)</p> <p>The default value of <i>n</i> is 25.</p>
VM_DUP_NUM_ACTIVE	<p>Specifies the maximum number of concurrent duplications that can run on a system. The default is 1.</p> <p>The maximum should be no greater than half the number of drives that are available to the duplication.</p>

Table C-1

Volume Management Configuration Parameters (Continued)

Parameter	Description
VM_DUP_STATE	Specifies whether media duplication is enabled or disabled. The default is enabled. This field appears only when its default value changes.
VM_DUP_TAPE_PAD	Specifies the amount of label padding applied to original volumes when duplication is enabled. The range is 1 to 10, where 1 is 10 MB of pad space and 10 is 100 MB of pad space. The default is 10. This field appears only when its default value changes.
LM_START : 0 LM_NAME: hp_c17xx_0 LM_HOST : edmdoc LM_EXEC_OPT : "" LM_EJECT_DEST : offline_0 LM_END : 0	Configuration parameters for each Library Manager configured for the server. All changes to this portion of vm.cfg are made by lmconfig. LM_NAME provides the name of the library unit in the form <i>manufacturer_library unit type_serial number</i> . LM_HOST specifies the name of the host on which the Library Manager daemon resides. The default is the localhost. LM_EXEC_OPT provides the options with which a particular Library Manager is started. The default is no value. The LM_EJECT_DEST specifies the name of the Library Manager that receives a volume when an eject occurs. The eject must always be initiated by clicking the Eject button in the EDM Library Unit Manager and not by the hardware eject button. The values are none, offline_0, and offsite_0.

Library Manager Configuration Files

The **lmconfig** utility creates a separate directory in `/usr/epoch/etc/lm` for each Library Manager that you configure. The name of each Library Manager follows a convention that is based on the type of library unit that it supports.

Library Manager Naming Convention

The Library Manager naming convention is of the form:

manufacturer_model_n

where:

<i>manufacturer</i>	Two- to three-character abbreviation that identifies the manufacturer of the library unit. For example, "hp" stands for Hewlett-Packard and "atl" stands for ATL Products.
<i>model</i>	Code (from two to five characters) that identifies the type of library unit. For example, <code>atl_452</code> supports the ACL 4/52 library units with DLT drives manufactured by ATL Products.
<i>n</i>	One-digit suffix that lmconfig appends to the Library Manager name. The suffix for the first Library Manager is 0. This number increments by 1 for each Library Manager of the same type that you configure. For example, the first Library Manager for an ATL 4/52 DTL library unit is <code>atl_452_0</code> and the second instance is <code>atl_452_1</code> .

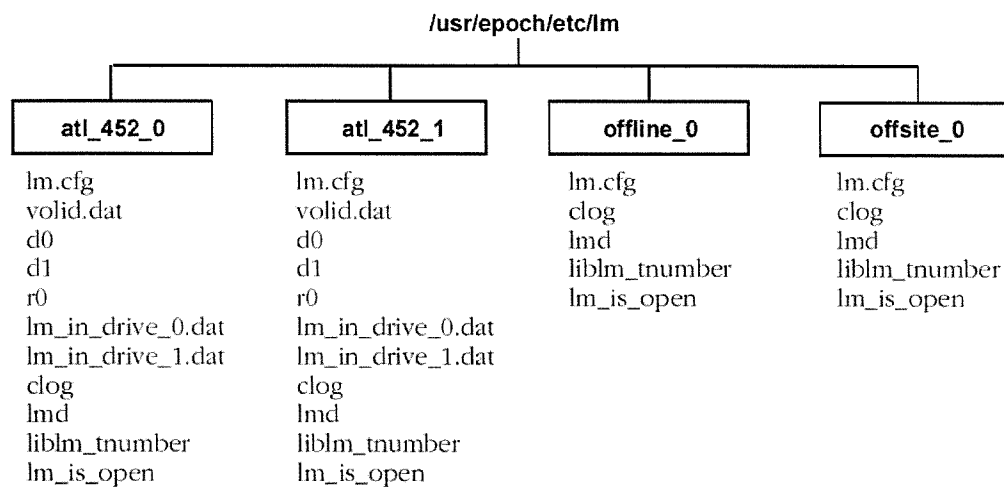
The **lmconfig** utility copies a template configuration file, `lm.cfg`, into the subdirectory and modifies the file based on input that you provide interactively. **lmconfig** also creates a link to the executable file and adds the information about the new Library Manager to the Volume Manager's configuration file.

Within each Library Manager directory, `lmconfig` creates several files. During start-up, the Library Manager reads the files in its directory to initialize the library unit that it is managing and to set up its internal data structures.

Figure C-2 on page C-10 illustrates an example of the Library Manager directory structure. (Note that though the library unit in the example contains four drives, only two appear in the example, due to space limitations.)

Refer to Table A-3 on page A-19 for a description of the files that reside in each Library Manager directory.

Sample Library Manager Directory Structure



Library Manager Configuration Parameters

Each Library Manager has a configuration file that contains one parameter and corresponding value per line. A delimiter separates each parameter from its value; at least one space appears before and after the delimiter as shown in the following example:

```
LM_NAME : atl_452_0
```

The configuration file contains parameters that define the name of the Library Manager, hardware addresses of the library unit and drives, and library unit operating features.

CAUTION: Do not edit the lm.cfg file manually. The lmconfig utility makes all modifications to Library Manager configuration files. If you need to make additional modifications to this file, contact EMC Customer Service for assistance.

Figure C-3 on page C-12 shows an example of the lm.cfg file. This is the default configuration file for the ATL 4/52 DLT library unit. (Note that comment lines begin with the # character.)

The lm.cfg file is set up during installation. When you change the hardware configuration by adding or removing a library unit, you need to reconfigure the software to recognize the change. For directions, refer to Chapter 17 “Configuring Library Managers”.

Table C-2 describes the Library Manager configuration parameters and default values in the lm.cfg file.

Figure C-3

Sample Library Manager Configuration File

```
###generated by lmconfig###
@@@begin_name@@@
LM_NAME : atl_452_0

@@@end_name@@@

LM_INLET : 0
    LM_INLET_STATE : enabled
LM_END_INLET : 0

LM_INLET : 1
    LM_INLET_STATE : enabled
LM_END_INLET : 1

LM_INLET : 2
    LM_INLET_STATE : enabled
LM_END_INLET : 2

LM_INLET : 3
    LM_INLET_STATE : enabled
LM_END_INLET : 3

@@@begin_other@@@
LM_VOLID_MAP : valid.dat
LM_DRIVE_EJECT_BEFORE_MOVE : 1
LM_BARCODE_CAPABLE : 1
LM_AUTO_INJECT : 1
LM_IES_ON_STARTUP : 0
LM_IES_ON_INVENTORY : 0

LM_SCAN_TIME : 5
LM_MAX_IDLE_TIME : 300
LM_MAX_RESIDENT_TIME : 7200
LM_MIN_RESIDENT_TIME : 120
LM_DRIVE_CLEAN_TIME : 400
```

```
LM_MOUNT_PRIORITY : 4
LM_DISMOUNT_PRIORITY : 2
LM_INJECT_PRIORITY : 3
LM_EJECT_PRIORITY : 5
LM_INVENTORY_PRIORITY : 12
LM_MOVE_MEDIUM_PRIORITY : 6
LM_WR_LBL_PRIORITY : 8
LM_RD_LBL_PRIORITY : 7

LM_LU_PHYSLOC : B1 lab
LM_CLEANER_BARCODE_RANGE : CLN*

####end_other###
LM_MEDIA_FORMAT : EDM
####begin_robot###
LM_LU : 0
LM_LU_BOARD : 3
LM_LU_BUS : 0
LM_LU_TARGET : 0
LM_LU_LUN : 0
LM_LU_PHYS_DEV : /usr/epoch/etc/lm/atl_452_0/r0
LM_END_LU : 0

LM_PICKER : 0
LM_PICKER_STATE : enabled
LM_END_PICKER : 0

####end_robot###
####begin_drive###
LM_DRIVE : 0
  LM_DRIVE_BOARD : 3
  LM_DRIVE_BUS : 0
  LM_DRIVE_TARGET : 1
  LM_DRIVE_LUN : 0
  LM_DRIVE_PHYS_DEV : /usr/epoch/etc/lm/atl_452_0/d0
  LM_MEDIA_TYPE : DLT
  LM_DRIVE_STATE : enabled
  LM_DRIVE_ATTR : lm_drive_dirty_query_able
  LM_IN_DRIVE_FILE : lm_in_drive_0.dat
LM_END_DRIVE : 0
```

```
#####end_drive###
#####begin_drive###
LM_DRIVE : 1
  LM_DRIVE_BOARD : 3
  LM_DRIVE_BUS : 0
  LM_DRIVE_TARGET : 2
  LM_DRIVE_LUN : 0
  LM_DRIVE_PHYS_DEV : /usr/epoch/etc/lm/at1_452_0/d1
  LM_MEDIA_TYPE : DLT
  LM_DRIVE_STATE : enabled
  LM_DRIVE_ATTR : lm_drive_dirty_query_able
  LM_IN_DRIVE_FILE : lm_in_drive_1.dat
LM_END_DRIVE : 1

#####end_drive###
#####begin_drive###
LM_DRIVE : 2
  LM_DRIVE_BOARD : 3
  LM_DRIVE_BUS : 0
  LM_DRIVE_TARGET : 3
  LM_DRIVE_LUN : 0
  LM_DRIVE_PHYS_DEV : /usr/epoch/etc/lm/at1_452_0/d2
  LM_MEDIA_TYPE : DLT
  LM_DRIVE_STATE : enabled
  LM_DRIVE_ATTR : lm_drive_dirty_query_able
  LM_IN_DRIVE_FILE : lm_in_drive_2.dat
LM_END_DRIVE : 2

#####end_drive###
#####begin_drive###
LM_DRIVE : 3
  LM_DRIVE_BOARD : 3
  LM_DRIVE_BUS : 0
  LM_DRIVE_TARGET : 4
  LM_DRIVE_LUN : 0
  LM_DRIVE_PHYS_DEV : /usr/epoch/etc/lm/at1_452_0/d3
  LM_MEDIA_TYPE : DLT
  LM_DRIVE_STATE : enabled
  LM_DRIVE_ATTR : lm_drive_dirty_query_able
  LM_IN_DRIVE_FILE : lm_in_drive_3.dat
LM_END_DRIVE : 3

#####end_drive###
###completed by lmconfig###
```

Table C-2

Library Manager Configuration Parameters

Parameter	Definition
LM_NAME : <i>name</i>	Name of the Library Manager. The LM_NAME appears next to the library icon in the Library Units and Drives area of the EDM Library Unit Manager window. The value is a character string that contains up to 16 characters.
LM_INLET : <i>n</i> LM_INLET_STATE : <i>state</i> LM_END_INLET : <i>n</i>	State of the inlet when the Library Manager starts up. The value is enabled or disabled. The relative inlet number is specified by LM_INLET and LM_END_INLET and must be the same value.
LM_VOLID_MAP : <i>file name</i>	Name of the file (volid.dat) that contains a table of the library unit's slot contents. The volid.dat file enables the Library Manager to start up without taking a complete inventory of the library unit.
LM_DRIVE_EJECT_BEFORE_MOVE : <i>n</i>	Determines whether hardware needs the volume to be ejected from the drive before the robot moves the volume. Values for <i>n</i> are 1 (enable, eject needed) or 0 (disable, eject is not needed).
LM_BARCODE_CAPABLE : <i>n</i>	Specifies whether the library unit supports barcodes. The value for <i>n</i> is 0 (no barcode support) or 1 (barcode support).
LM_AUTO_INJECT : <i>n</i>	Configures the library unit's inlet as automatic or manual. Values are: 0 = manual inlet and 1 = automatic inlet. A manual inlet (that is, hardware-controlled) requires the user to click the Inject button before the robot moves the volume from the inlet into the library unit. If the inlet is configured as automatic, the Library Manager polls the inlet and moves the volume into the library units without an explicit command. The time interval in which the inlet is polled is specified by the LM_SCAN_TIME parameter.
LM_NO_IES_ON_STARTUP : <i>n</i> OR LM_IES_ON_STARTUP : <i>n</i>	Determines whether hardware checking occurs (e.g., obtaining the status of drives, slots, and inlets) at startup and during inventory.
LM_NO_IES_ON_INVENTORY : <i>n</i> OR LM_IES_ON_INVENTORY : <i>n</i>	If the values of the parameters that contain "NO" are set to 1 or the values of the parameters that imply "YES" are set to 0, no hardware checking occurs.

Table C-2

Library Manager Configuration Parameters (Continued)

Parameter	Definition
LM_SCAN_TIME : <i>n</i>	Time interval (in seconds) that specifies how often the Library Manager polls its work list (which includes polling inlets, new requests, and cancellations); the default is 5 seconds . For automatic inlets, the inlet is polled continuously on this interval. For manual inlets, the inlet is polled when you click the Inject button.
LM_MAX_IDLE_TIME : <i>n</i>	Maximum time (in seconds) that a volume can remain in the drive after a dismount request is made. The default value for all drives is 300 seconds (five minutes).
LM_MAX_RESIDENT_TIME : <i>n</i>	Maximum time (in seconds) that a volume can remain in a drive before allowing preemption by a volume of the same priority. The default value for EO drives is 120 seconds (two minutes). The default value for tape drives is 7200 seconds (two hours).
LM_MIN_RESIDENT_TIME : <i>n</i>	Minimum time (in seconds) that a volume of a lower priority can be in a drive before allowing preemption for a volume of a higher priority. The default value is 120 seconds for tape (two minutes), 30 seconds for optical media.
LM_DRIVE_CLEAN_TIME : <i>n</i>	Designated time (in seconds) in which to clean a drive. The Library Manager starts verifying whether the cleaning completed after this time period elapsed. The default value is 300 seconds (5 minutes) or 400 seconds (over six minutes), depending on the type of drive and library unit.
LM_MOUNT_PRIORITY : <i>n</i>	Default priority for mount operations. The value is an integer in the range 1-16 (the default is 4) where 1 represents the highest priority.
LM_DISMOUNT_PRIORITY : <i>n</i>	Default priority for dismount operations. The value is an integer in the range 1-16 (the default is 2); where 1 represents the highest priority.
LM_INJECT_PRIORITY : <i>n</i>	Default priority for inject operations. The value is an integer in the range 1-16 (the default is 3); where 1 represents the highest priority.
LM_EJECT_PRIORITY : <i>n</i>	Default priority for eject operations. The value is an integer in the range 1-16 (the default is 5); where 1 represents the highest priority.
LM_INVENTORY_PRIORITY : <i>n</i>	Default priority for movement operations. The value is an integer in the range 1-16 (the default is 12); where 1 represents the highest priority.

Table C-2

Library Manager Configuration Parameters (Continued)

Parameter	Definition
LM_MOVE_MEDIUM_PRIORITY : <i>n</i>	Default priority for label write operations. The value is an integer in the range 1-16 (the default is 6); where 1 represents the highest priority.
LM_WR_LBL_PRIORITY : <i>n</i>	Default priority for label read operations. The value is an integer in the range 1-16 (the default is 8); where 1 represents the highest priority.
LM_RD_LBL_PRIORITY : <i>n</i>	Default priority for inventory operations. The value is an integer in the range 1-16 (the default is 7); where 1 represents the highest priority.
LM_LU_PHYSLOC : <i>string</i>	Physical location of the library unit, which the user enters while running lmconfig.
LU_CLENER_BARCODE_RANGE : <i>string</i>	Cleaner barcode range. If the value is "N/A" or this parameter does not appear in the lm.cfg file, no assumptions are made for cleaner barcodes. If the value is "CLN*," any barcode that begins with "CLN" identifies a cleaner.
LM_MEDIA_FORMAT : <i>string</i>	Format of the backup data on the tape. The supported value is "EDM."
LM_LU : <i>n</i> LM_LU_BOARD : <i>n</i> LM_LU_BUS : <i>n</i> LM_LU_TARGET : <i>n</i> LM_LU_LUN : <i>n</i> LM_LU_PHYS_DEV : <i>path</i> LM_END_LU : <i>n</i>	Parameters between LM_LU and LM_END_LU (except for LM_LU_PHYS_DEV) that define the system board number, I/O bus slot number, SCSI target ID, logical unit number (LUN). All values are integers. LM_LU and LM_END_LU represent the relative library unit number and must be the same value. LM_LU_PHYS_DEV provides the full pathname (in a character string) of the physical device node for the library unit.
LM_PICKER : <i>n</i> LM_PICKER_STATE : <i>state</i> LM_END_PICKER : <i>n</i>	State of the robot (or robot) when the Library Manager starts up. The value is enabled or disabled. The relative robot number is specified by LM_PICKER and LM_END_PICKER and must be the same value.
LM_BARCODE_VERIFICATION : <i>value</i>	Specifies whether the barcode should be verified when a volume is mounted. The value is ignored.

Table C-2

Library Manager Configuration Parameters (Continued)

Parameter	Definition
LM_DRIVE : <i>n</i> LM_DRIVE_BOARD : <i>n</i> LM_DRIVE_BUS : <i>n</i> LM_DRIVE_TARGET : <i>n</i> LM_DRIVE_LUN : <i>n</i> LM_DRIVE_PHYS_DEV : <i>path</i> LM_MEDIA_TYPE : <i>value</i> LM_DRIVE_STATE : <i>value</i> LM_DRIVE_ATTR : <i>string</i> LM_IN_DRIVE_FILE : <i>file name</i> LM_END_DRIVE : <i>n</i>	LM_DRIVE and LM_END_DRIVE represent the relative drive number and must be the same value. LM_DRIVE_BOARD defines the system board number. LM_DRIVE_BUS defines the I/O bus slot number. LM_DRIVE_TARGET is the SCSI target ID. LM_DRIVE_LUN is the logical unit number (LUN). Other parameters are described below.
LM_DRIVE_PHYS_DEV : <i>path</i>	Provides the full pathname (in a character string) of the physical device node for the drive or robot.
LM_MEDIA_TYPE : <i>value</i>	Type of media that the drive supports. To configure a multifunction drive, specify one LM_MEDIA_TYPE parameter and value for each media type: DLT = digital linear tape cartridge DTF = digital tape format HITC_STK, HITC_MAGSTAR = half-inch tape cartridge EO = erasable optical disk cartridge WORM = write once, read many optical disk cartridge 9840s = ?
LM_DRIVE_STATE : <i>value</i>	State of the drive when the Library Manager starts up. The <i>value</i> is enabled or disabled.
LM_DRIVE_ATTR : <i>string</i>	Indicates whether EDM is to determine whether a drive is dirty. If this parameter is not set, EDM does not check the drives; you must then check the drives directly.
LM_IN_DRIVE_FILE : <i>name</i>	File name of the file that contains information about the contents of the drive. The default is <i>lm_in_drive_n.dat</i> , where <i>n</i> is the relative drive number. The drive content file is used during start up and is updated by the Library Manager each time a volume is moved into and out of a drive. If a drive content file does not exist for each drive, the Library Manager creates one during startup time.

Table C-2

Library Manager Configuration Parameters (Continued)

Parameter	Definition
LM_DRIVE_START_AFTER_MOVE : <i>n</i>	<p>Indicates whether the drive needs to be explicitly started after a volume is mounted.</p> <p>The default value is 0, no explicit start needed. A value of 1 indicates explicit start.</p>
LM_INJECT_TIMEOUT : <i>n</i>	<p>Maximum time (in seconds) that the Library Manager waits for a volume to be inserted into the inlet before returning a timeout error. This parameter is used only for library units that have software control, which LM_INLET_UNLOCK_NEEDED specifies.</p>
LM_ATTR: <i>value</i>	<p>Defines the Library Manager; <i>value</i> is offline or offsite. This parameter applies only to Library Managers without physical library units.</p>
LM_OFFLINE_MOUNT_ACTION : <i>value</i>	<p>Defines the mount action to be taken upon receipt of a volume mount request. The value is: error, which automatically rejects the request, or queue, which holds the request in a queue until the request is satisfied or manually rejected.</p> <p>This parameter applies only to Library Managers without physical library units.</p>
LM_OFFLINE_MOUNT_USR_ROUTINE : <i>path</i>	<p>Specifies the pathname of a user routine that defines the mount action. For example, a user routine could be written to send a mail message to the operator when a volume request is made for an offline volume.</p> <p>This parameter applies only to Library Managers without physical library units.</p>

**The LM_INLET_IGNORE_ON_OPEN
Parameter**

The LM_INLET_IGNORE_ON_OPEN volume enables you to re-inject volumes that you just ejected into the library unit automatically, by just opening and closing the inlet. The default for this capability is 0:

LM_INLET_IGNORE_ON_OPEN : 0

When set to the default, this parameter does not appear in the lm.cfg file. However, you can set the parameter to 1, which removes this automatic re-injection feature. You must then remove media from the inlet and then close the inlet before ejected volumes can be re-injected.

To set the parameter to 1, add this parameter to the lm.cfg file manually, anywhere between "begin_other" and "end_other." For example:

```
#####begin_other###
LM_VOLID_MAP : valid.dat
LM_DRIVE_EJECT_BEFORE_MOVE : 1
LM_BARCODE_CAPABLE : 1
LM_AUTO_INJECT : 1
LM_IES_ON_STARTUP : 0
LM_IES_ON_INVENTORY : 0
LM_INLET_IGNORE_ON_OPEN : 1
.
.
.
#####end_other###
```

Offline and Offsite Library Managers

The offline and offsite Library Managers each has its own configuration file. Each Library Manager is described below.

Offline Library Manager

The offline Library Manager resides in /usr/epoch/etc/lm/offline_0. Figure C-4 shows the parameters that its configuration file contains. Refer to Table C-2 on page C-15 for a description of configuration parameters.

Figure C-4**Offline LM Configuration File**

```

###generated by lmconfig###
####begin_name###
LM_NAME: offline_0

####end_name###
####begin_other###
LM_SCAN_TIME: 5
LM_OFFLINE_MOUNT_ACTION: queue
LM_OFFLINE_MOUNT_USR_ROUTINE: /bin/true

####end_other###
###completed by lmconfig###

```

Offsite Library Manager

The offsite Library Manager resides in /usr/epoch/etc/lm/offsite_0. Figure C-5 shows a sample configuration file. The parameter LM_ATTR indicates that the Library Manager is offsite, as opposed to offline. The parameter LM_OFFLINE_MOUNT_ACTION is described in Table C-2.

Figure C-5**Offsite LM Configuration File**

```

###generated by lmconfig###
####begin_name###
LM_NAME : offsite_0

####end_name###
####begin_other###

LM_ATTR : offsite
LM_OFFLINE_MOUNT_ACTION: error

####end_other###
###completed by lmconfig###

```

D findxcpio Directives

The work item directive specifies which filesystems, directories, and files to back up on the client, and which to exclude. EDM Backup's autoconfiguration feature builds these directives for you automatically.

When backups are run, all directives, which can consist of macros, are translated into the expanded **findxcpio** syntax; the **findxcpio** program scans the client filesystem and reads the client data. (Refer to the **findxcpio** man page for detailed information about the command.)

This chapter explains the **findxcpio** macros and expanded syntax.

- Work Item Directive
- Logical Operators
- Macros
- Syntax to do Back Up
- Syntax to not Back Up
- Evaluation Shortcuts

Work Item Directive

You can use the Work Item tab in the EDM Backup Configuration window to adjust the specification for the directive(s). The interface builds the directives for you (with macros). In addition, you can also directly specify the **findxcpio** directive in the Work Item Options window which is accessed through the Work Item tab in the EDM Backup Configuration window. (You can also edit the directives in the eb.cfg configuration file.)

The general format of the work item directive is:

```
work item: "work item name", "client name"
{
  "files to back up";
  "other statements";
}
```

This statement has three fields:

- *Work item name*: specifies a unique work item that indicates the client's name and perhaps a set of files to be backed up.
- *Client name*: specifies the client name.
- *Files to back up*: specifies the client's files that you want to back up. Certain **findxcpio** qualifiers can be used after the specifying file names, to further refine which files are to be backed up.

Logical Operators

The **findxcpio** qualifiers include logical operators that enable you to expand the file specifications with evaluation statements. These operators are:

- **-o** (or)
- **!** (not)
- **-a** (and)

The **-a** operator is not required; it is implied when two or more evaluation statements occur in a row.

Macros

EDM Backup provides a set of macros to use in place of the **findxcpio** syntax. The macros simplify specifying what files to back up and which to skip. They are defined in the **startfind** script.

The **findxcpio** command uses arguments that are similar to the Unix **find** command. Therefore, all “DO...” macros must be positioned before other macros, such as “PRUNE...” macros.

Note: Simpler representations of these macros are available in the Work Item tab of the EDM Backup Configuration window.

Table D-1 lists basic macros and their expanded **findxcpio** syntax. In these statements, substitute the name of a filesystem for *fs*.

Table D-1

Basic findxcpio Macros

This Macro...	Expands to this findxcpio Syntax
ROOTDIR	"/"
DO_FS <i>fs</i>	_fs
DO_PATH <i>fs</i>	_fs
DO_DIR <i>fs</i>	_fs
DO_FILE <i>fs</i>	_fs
PRUNE_PATH <i>fs</i>	"\ (-path", _fs, "-prune -o -true \)"
PRUNE_DIR <i>fs</i>	"\ (\! -name", _fs, "-prune -o -true \)"
PRUNE_FILE <i>fs</i>	"\ (\! -name", _fs, "-prune -o -true \)"
LOCAL_FS_ONLY	"-xdev"
NO_NFS	"\ (-fstype nfs -prune -o -true \)"

Each of the following examples shows how to use the macros that are listed in Table D-1 in a work item directive:

- To back up the root (/) filesystem, you enter the work item:
`work item: "cad1-all", "cad1", ROOTDIR;`
- To back up the /home filesystem and all files and directories under it, you enter the work item:
`work item: "cad1-all", "cad1",
DO_PATH("/home");`
- To skip any files in /tmp, you enter the work item:
`work item: "cad1-all", "cad1",
PRUNE_PATH("/tmp");`
- To back up all filesystems in the root (/) filesystem and to skip any NFS files, you enter the work item:
`work item: "cad1-all", "cad1",
DO_PATH("/"), NO_NFS;`
- To back up all filesystems in the root (/) filesystem, and to skip any files named core, you enter the work item:
`work item: "cad1-all", "cad1",
DO_FS("/"), PRUNE_FILE("core");`
- To back up only files in a particular filesystem (for example, /usr), you enter the work item:
`work item: "cad1-all", "cad1",
DO_FS("/usr"), LOCAL_FS_ONLY;`

Compound Macros

Combining the macros in Table D-1, EDM Backup provides the following compound macros.

Table D-2

Compound findxcpio Macros (1)

This Macro...	Expands to this findxcpio Syntax
NO_TMP_FILES	PRUNE_PATH ("/tmp"), PRUNE_FILE ("*~"), PRUNE_FILE ("\#*")
NO_AUTOMOUNT	PRUNE_PATH ("/net"), PRUNE_PATH ("/tmp_mnt"), PRUNE_PATH ("/nfs")

Each of the following examples shows how to use the macros that are listed in Table D-2 in a work item directive:

- To back up files in the root (/) filesystem, and to skip files in the temporary directory (/tmp), files that end with a tilde (~), and files that start with a pound sign (#), you enter the work item:

```
work item: "cad1-all", "cad1",
DO_PATH ("/"),NO_TMP_FILES;
```
- To back up files in the root (/) filesystem, and to skip any files in the /net, /tmp_mnt, or /nfs directories, you enter the work item:

```
work item: "cad1-all", "cad1",
DO_PATH ("/"),NO_AUTOMOUNT;
```

EDM Backup also provides additional compound macros that combine the macros that Table D-1 and Table D-2 contain. The following table lists the five compound macros and shows their expanded **findxcpio** syntax.

Table D-3**Compound findxcpio Macros (2)**

This Macro...	Expands to this findxcpio Syntax
ROOT_ONLY	DO_PATH("/"), LOCAL_FS_ONLY
LOCAL_DISK	DO_PATH("/"), NO_AUTOMOUNT, NO_NFS
SYSTEM_FILES	DO_PATH("/usr"), NO_AUTOMOUNT, LOCAL_FS_ONLY
USER_FILES	DO_PATH("/home"), NO_TMP_FILES, LOCAL_FS_ONLY
LOCAL_FILES	DO_PATH("/"), NO_NFS

Each of the following examples shows how to use the macros that are listed in Table D-3 in a work item directive:

- To back up only local files in the root (/) filesystem, you enter the work item:
work item: "cad1-all", "cad1", ROOT_ONLY;
- To back up all filesystems on a client's local disk, and to skip backing up any automounted or NFS filesystems, you enter the work item:
work item: "cad1-all", "cad1", LOCAL_DISK;
- To back up two local filesystems, root (/) and /usr, and to skip any NFS-mounted filesystems, you enter the work item:
work item: "cad1-all", "cad1", SYSTEM_FILES;
- To back up only the local filesystem and to skip any temporary files, you enter the work item:
work item: "cad1-all", "cad1", USER_FILES;
- To back up all filesystems on the local disk, and to skip any NFS filesystems, you enter the work item:

```
work item: "cad1-all", "cad1", LOCAL_FILES;
```

Syntax to do Back Up

The **findxcpio** command options makes it easy to tell EDM Backup to search for certain types of files for backup.

When specifying a client backup list, the simplest backup file specification is "/" – which means to start at the top of the filesystem and back up all files and directories. In addition to "/" you can specify other qualifiers to select or deselect files.

For example, to specify to have the backup program back up only files it finds under "/" that are .c files, you use the **-name** option:

```
work item: "cad1-all", "cad1", "/" -name '*.c' ;
```

This file specification checks for all .c files. When EDM Backup locates .c files, it backs them up. When EDM Backup finds files that are not .c files, it passes over them. Note the use of single quotes around the *.c expression. The quotes are required because the asterisk (*) is a special character to the shell and the quotes prevent the * from being expanded prematurely.

Suppose you want EDM Backup to copy the files that are stored on a fileserver that belong to a particular user. You use the **-user** option with the name of a user (karen):

```
work item: "atlas1", "disk1-all", "/" -user karen;
```

This file specification checks for all files under the / directory that karen owns, and backs them up. EDM Backup passes over all files that the user karen does not own.

Similarly, to select files that belong to a particular group, you use the **-group** option. To back up all files that the doc group owns on the fileserver atlas1, you enter:

```
work item: "atlas1", "disk2-all", "/" -group doc;
```

This file specification checks for all files under the “/” directory owned by members of the doc group, and backs them up. EDM Backup passes over all files that members of group doc do not own.

To select files for backup that contain a certain number of blocks and that have been accessed within a certain number of days, use the **-size** and **-atime** options. For example, to select files that contain 30,000 blocks (512 bytes each) and were accessed in the last 72 hours, you enter:

```
work item: "cad1-all", "cad1", "/" -size 30000 -  
atime -3";
```

This file specification checks for files under the “/” directory that contain 30,000 blocks and were accessed within the last 72 hours (specified by the value -3 because each 24-hour period is represented by a value of 1) and backs them up. EDM Backup passes over any other files.

To select a certain type of file for backup that was changed within a certain number of hours, use the **-type** option and the **-ctime** option. For example, to select symbolic links (**l**) that were modified within the last 48 hours, you enter:

```
work item: "cad1-all", "cad1", "/" -type l -ctime -  
2";
```

This file specification searches for all symbolic links that were modified in the past 48 hours (specified by the -2 because each 24-hour period is represented by a 1) and backs them up, passing over any other files.

On an HSM system, to select only staged files (files with a staging ID #1 filled in), use the **-staged** option:

```
work item: "cad1-all", "cad1", "/" -staged";
```

Syntax to not Back Up

The **findxcpio** command options make it easy to tell EDM Backup to exclude certain types of files from the backup list.

For example, if you want to add a check so that EDM Backup does not back up a client's NFS filesystems, you add the evaluation for the filesystem type NFS using the **-fstype** and **-prune** options. The **-fstype** option directs EDM Backup to search for a particular filesystem type. The **-prune** option directs EDM Backup not to descend into subdirectories, but continue evaluating other filesystems for backup. To direct EDM Backup not to copy NFS filesystems under the "/" directory you enter:

```
work item: "cad1-all", "cad1", "/ \( -fstype nfs -prune -o -true \)";
```

Note: The use of **-fstype nfs -prune** is not recommended, because it is impossible to prevent an occasional descent into an NFS filesystem. The better approach is to specify individual filesystems, each with the **-xdev** option.

This file specification checks for NFS filesystems, and when EDM Backup finds an NFS filesystem (*true*) then the evaluation continues to the second statement, which is **-prune**. **-prune** always evaluates to *true* and causes the search to stop at the NFS filesystem mount point, and to skip over it. If the files are not NFS (*false*), the evaluation proceeds to the second statement (**-o -true**), which evaluates to *true* and directs EDM Backup to copy the files.

If your site uses automount, and you find that you have timeout problems using the previous file specification, use this syntax:

```
work item: "cad1-all", "cad1", "/ \( -path /net -prune -o -path /nfs -prune -o -path /tmp_mnt -prune \) -o -true";
```

Note: Although this line appears as multiple lines in this example, it must be specified on one line in the configuration file.

This file specification checks for files in three paths (in this example, /net, /nfs, and /tmp_mnt). When EDM Backup finds one of these paths (*true*) then the evaluation continues to the second statement (**-prune**). **-prune** always evaluates to *true* and causes the search to stop at the specified path and skip over it.

Suppose you do not want to back up a client's /tmp and /usr/tmp directories in the "/" directory. To add checks for these directories you use the **-path** option to specify the pathnames to exclude. You specify the **-o** option (meaning *or*) to direct EDM Backup to exclude either directory:

```
work item: "cad1-all", "cad1", "/" ( -path /tmp -o -path /usr/tmp \) -prune -o -
true";
```

This file specification verifies whether the directory is /tmp or /usr/tmp. If the directory is neither (*false*) the evaluation stops and EDM Backup copies the file to the backup. If the directory is /tmp or /usr/tmp (*true*) then the evaluation goes to the second statement **-prune**, which always evaluates to *true* and causes the search to stop at the /tmp or the /usr/tmp directory.

Suppose that in addition to not backing up the /tmp or /usr/tmp directories, you do not want to back up editor files and core files. To specify this, you use the **-name** option to add checks for editor files (*~) or core files:

```
work item: "cad1-all", "cad1", "/" (-path /tmp -o -path /usr/tmp \) -prune -o ! (-
name '*~' -o -name core)";
```

The first part of the file specification, as explained in the previous example, checks whether the directory is /tmp or /usr/tmp, and does not copy the contents of the directory if it is either directory. The second part of the file specification checks for files that have the name *~ or core (**-name**). This statement instructs EDM Backup that if the files are not editor backup files (*~) or core files (*true*) then back them up. If the files are one of these (*false*), do not back them up.

Suppose you want to instruct EDM Backup not to cross filesystem boundaries, which excludes from backup any filesystems that are not explicitly listed. You can use the **-xdev** option for this purpose:

```
work item: "atlas1", "disk1-all", "/" /usr /homes -
xdev";
```

Note: If NFS timeouts cause backups to fail, try using the **-xdev** switch.

This file specification checks for the specified filesystems (/ , /usr, and /homes) and instruct EDM Backup not to cross filesystem boundaries, and so, to only back up the listed filesystems. Because the file specification includes the "/" filesystem, if the **-xdev** switch is not included, EDM Backup backs up everything in the namespace, including NFS-mounted filesystems.

Evaluation Shortcuts

In certain cases, the **findxcpio** evaluation process can skip parts of the file specification in the work item directive, saving time and preventing unwanted side-effects. The **findxcpio** program saves time by quickly eliminating directories and files that are specified for exclusion from the backup list. In doing so, the program avoids unwanted side-effects because it does not act upon the second half of the expression.

Here are the two cases when **findxcpio** does not bother with the second half of a directive.

False and

In directives with an *and* in the evaluation, if the first statement evaluates to *false*, **findxcpio** short circuits the evaluation and ignores the second half of the directive (**-prune**).

Consider this work item directive that specifies not to back up a client's /tmp and /usr/tmp directories:

```
work item: "cad1-all", "cad1", "/ \(-path /tmp -o -path /usr/tmp\) -prune o -true";
```

In the first part of this directive that contains the implied *and*, when **findxcpio** finds that a directory is not /tmp or /usr/tmp (*false*) then **findxcpio** short circuits and does not evaluate the **-prune** option. On the other hand, if the directory is /tmp or /usr/tmp (*true*) then **findxcpio** evaluates the **-prune** option. (The **-o -true** statement directs EDM Backup to copy files that are not /tmp or /usr/tmp.)

True or

In directives with an *or* evaluation (**-o**), if the first statement evaluates to *true*, **findxcpio** short circuits the evaluation and does not evaluate the second half of the directive. Thus, in the following example, if the directory is */tmp* (*true*) then **findxcpio** does not check for the second type of files (*/usr/tmp*), but goes directly to the **-prune** option:

```
work item: "cad1-all", "cad1", "/ \(-path /tmp -o -path /usr/tmp\) -prune";
```

Glossary

Access Control List (ACL)	Provides an enhanced level of security for UNIX files. An ACL extends the standard UNIX permission settings beyond owner, group, and other. An owner of a file can permit or deny access to specific users and groups.
allocation request	Request for a volume sent by an application. The volume characteristics are specified in the accompanying volume template.
archive setting	Sample HSM watermark setting that is intended for filesystems whose files are written once and rarely, if ever, read. The filesystem's data will typically be staged-out and rarely, if ever, staged back in. This would be the case if large amounts of data are gathered every day and quickly "archived" off of the magnetic disk. Other sample watermark settings include cached setting and random setting.
autochanger	Robotic mechanism inside a library unit that physically moves media into and out of slots and drives. Sometimes referred to as a "picker" or "robot."

automatic scheduling	Function that automatically schedules incremental backups as well as some full backups in order to back up each work item each night. It provides a full backup of each work item once within the rotation period. It attempts to distribute the work so that each night's backups will operate for approximately the same length of time. See also custom scheduling.
backup activity monitoring	Allows viewing and management of active, successful, and failed work items through the EDM graphical user interface.
Backup Activity Wizard	Enables you to start new, queued, or failed backups, stop running backups, or manage the backup queue. Access this wizard from the EDM Main window.
backup catalog	Group of related files that maintain a continuing backup history. A catalog identifies a backup at the file level by recording the names and attributes of each file on the client system at the time of the backup. Backup catalogs also keep track of the location of backup data for each file that was selected for backup.
backup catalog delta	Contains a condensed backup catalog with information that differs only from the previous backup catalog files.
backup configuration	Set of parameters on the backup server used to define what data gets backed up, when it gets backed up, to where it is backed up, how backups are processed, and who can run backups and restores. These parameters are edited through the Backup Configuration window and stored in the /usr/epoch/EB/config/eb.cfg file.
Backup Configuration window	Window in the EDM graphical user interface for editing the backup configuration.

Backup Configuration Wizard	Enables you to configure a network, Symmetrix Path, or Symmetrix Connect backup of filesystems or a database. It supports all clients. You access this wizard through the Main window of the EDM GUI. It leads you step-by-step through the configuration process.
backup levels 1–9	Specifies a backup in which EDM Backup copies only those files that changed since the last backup of a lower level.
backup media	Media for storing backup data. Parameters for the backup media are specified in the trailset and trail of the backup configuration.
backup saveset	EDM Backup creates a saveset record for each work item it backs up. A saveset record contains the template name, work item name, the backup level, start and completion times, expiration times, and the backup trail. The saveset record is used to find the volume containing the backup data and the associated backup catalog.
backup saveset record	Data saved on backup media from a single backup of a single work item.
backup schedule template	Specifies all the information about how to perform a backup. It includes the work group(s) to backup, the rotation period and backup shift lengths, the trailsets on which to store the backup data, and the backup schedule that dictates a backup level for each day. See also template.
backup server	EDM server on a network that contains the client, work item, media, schedule, and server configuration information.
backup shift	Parameter in the template of the backup configuration. It specifies the desired amount of time for backups to run in each 24-hour period.

- backup trailset** Also called media set. Defines the media trails to which the backup data is sent and the type of media to use. It also defines how long to save the backup data and its associated elements. See also trailset.
- bad file** File that changed during backup or a file that is corrupt. EDM Backup tries three times before marking a file bad.
- baseline backup** With baseline backups, you back up all of your most stable files, which, at minimum, consist of all the files that are staged out to the staging media. From that point on, you perform backups relative to the baseline; that is, the baseline backups take care of the data that is staged out, while the regular backups take care of everything else.
- bitfile** Uninterpreted stream of bytes that contains the staged-out portion of a file. A single bitfile can hold the contents of a single client file or the contents of multiple files. A bitfile is uniquely identified by a *bitfile ID* and a *store ID*.
- bucket** When EDM Migration stages out a file, it logically divides the file into a number of segments known as buckets, which can then be individually accessed.
- bulk staging** In HSM, bulk staging reduces each filesystem's magnetic disk utilization to a predefined low watermark (LWM). Bulk staging is another name for periodic staging.
- cached setting** Sample HSM watermark setting intended for filesystems in which reads outnumber writes, and a relatively predictable set of files are read. This setting takes advantage of migration's ability to keep the most recently-accessed files on magnetic disk, thus ensuring optimal performance. Other sample watermark settings include archive setting and random setting.

catalog	See backup catalog and volume catalog.
cataloged backup	Method of backup that updates the system administration database with information about the backup (i.e., the name of the template and the volume ID of the backup volumes).
client	Workstation or fileserver on a network that accesses the EDM server to back up and restore filesystem and database data, and optionally, migrate files.
client software	See local client software and remote client software.
client store	In HSM, a collection of files that have migrated from a single network client to the EDM server. The client store can reside within any stageable filesystem on the EDM fileserver. Every client store is associated with a <i>store ID</i> and a <i>bitfile ID</i> .
compaction	In HSM, the process of eliminating stale space on volumes in a staging trail and consolidating the staged files onto the minimum number of necessary volumes. Compacted volumes can then be erased and reused.
configuration	See backup configuration.
cross-client restore	User-initiated restore of their own backed up files on one client to their own directory on a different client.
custom scheduling	Explicitly schedule backups for particular days of the week, month, or other schedule period. (As opposed to using automatic scheduling, which schedules backups using general parameters and processing algorithms.)

data access pattern	Refers to the frequency in which staged data is accessed. Data access patterns include three categories: archive, random-retrieve, and general purpose.
database work item	Specifies which client databases you want to back up. You can specify which filesystems, directories, files, and raw partitions to include or exclude for backup. See also work item.
day of rotation	Rotation option within custom schedule that lets you schedule a period that is not a multiple of 7 days.
demand staging	Stage-out that occurs when the high watermark is reached.
device node	Special file, located in /dev, that acts as a pointer to a device driver. It associates a location, type, and access mode with a physical device.
disaster recovery	Recovery procedure for when the backup server's own disks crash. Also applies to crashed disks on network clients. In both cases, some EDM software might have been lost, requiring extra work before performing the data recovery with EDM Backup Restore window.
domain	In the GUI, reports can be designed and run for the local EDM or for an EDM domain of multiple EDMs. A domain consists of a domain master EDM machine and multiple EDM machines who agree to participate.
EDM (EMC Data Manager)	EMC's hardware product for use as the backup server. Provides network backup and restore with automated management of media. Contains the EDM Backup software and optional HSM software.

EDM Backup	EMC's software module for network backup and restore. Its interfaces include the EDM Backup Configuration window and the EDM Restore window.
EDM Backup client	Workstation or filesystem on a network that accesses the server to backup and restore files. See also remote client software.
EDM HSM	EMC's distributed hierarchical storage management application. The product consists of several software modules that support local and network migration.
EDM Migration	EDM Migration provides migration services between an EDM server and peripheral devices, such as optical or tape library units. It also provides HSM services for other file servers and workstations on the network.
EDM transfer protocol	The connection method, under control of the edmlinkd daemon, used by EDM for installing and communicating with supported clients. On supported clients (see the current <i>EDM Release Notes</i>) this replaces use of remote shell.
EDM Volume Management	Underlying software of the EDM Library Unit Manager interface that manages volume allocation, drive scheduling, and tracks all removable media for EDM Backup and optional HSM applications. See also Volume Manager and Library Unit Manager.
EMC Data Manager (EDM)	EMC's hardware product for use as the backup server. Provides network backup and restore with automated management of media. Contains the EDM Backup software and optional HSM software.
emxattr file	HSM extended attribute file. A file used by HSM that contains information about files that have been staged out.

erasable optical (EO) disk	Rewritable storage medium that uses laser technology to write data onto the disk. Also referred to as a magneto-optical disk.
event-driven staging	Method of staging that reduces a filesystem's magnetic disk utilization to a predefined low watermark. Event-driven staging occurs automatically when a filesystem reaches a predefined high watermark (HWM).
expire backups	Process that enables old data stored on backup media to be overwritten with new data, which includes deleting online catalogs from the backup server's disks.
expire catalogs	Process that deletes online catalogs from the backup server's disks, which can be done with or without expiring backups.
expired media	"Expired" is the media state that signifies a piece of media has exceeded its maximum number of uses.
explicit staging	Method of staging that is initiated manually by users who want to selectively stage out one or more files as a group.
fencepost	Portion of a file that remains on magnetic disk after the first stage out.
file restore	Copies a client's backup files from the backup server's media to the client's disk.
filesystem	Contains the files and directories on each individual disk partition. The "filesystem" refers to the overall system directory tree that merges these filesystems into a single hierarchy.

filesystem backup	Backup of filesystems over the network, using core EDM Backup functionality. Backup of data designated by filesystem work items.
filesystem work items	Defined unit of data to be backed up, consisting of one or more filesystems. Each work item is uniquely named and specifies the filesystems to be backed up.
full backup	Copies all the scanned <i>client</i> files, independent of the time of their last backup or their location, to the <i>backup server</i> .
green zone	Disk utilization level that is between the low and high water-marks. The green zone should be large enough to hold the average number of disk blocks used in a day including both new files and previously inactive (staged-out) files that are accessed (staged-in).
Hierarchical Storage Management (HSM)	Collection of techniques used to effectively manage a hierarchy of storage media such as RAM, magnetic disks, optical disk and tape. HSM uses techniques such as attempting to keep the most frequently accessed data on the highest speed media (highest speed usually implies highest cost), less frequently accessed data on the next highest speed media, continuing with this model until the least frequently accessed data is on the lowest speed and cost media.
high watermark (HWM)	Preconfigured disk utilization level that, when reached, causes HSM to immediately stage out enough files to secondary storage to reach the low watermark (LWM).
import	Method of moving volumes among servers. The volume management import process reads the electronic volume label and adds it to the volume catalog of the receiving server.

incremental backup	Backup method that copies only those client files that have changed since the previous backup of any level.
inode	UNIX file's directory information, for example its attributes or meta-data.
keyboard focus	Indicates the window or element within a window that receives keyboard input.
labeled volume	Media that has been given a volume label.
library unit	Robotic library unit that automatically manages the placement of cartridges. Most library units are equipped with an inlet to insert and eject media, robotics to physically move media, one or more internal drives, internal storage slots, and in some models, a barcode scanner.
Library Unit Manager	Process that manages volumes located in physical library units and offline and offsite locations. A Library Unit Manager manages drive scheduling, volume mounts and dismounts, volume injects and ejects, and library unit inventories. See also <code>offline_0</code> and <code>offsite_0</code> .
Library Unit Manager window	Part of the EDM graphical user interface that allows administrators to label, allocate, and acquire information about all volumes in use for backup and, optionally, HSM. Started from the EDM Main window.
life cycle of media	See volume life cycle.
local client software	Software located on the backup server that scans its disks and copies the data for backup. See also remote client software and server software.

local migration	Staging of filesystems from the EDM server to a tape or optical library. See also network migration client.
logical data	Data that is identified either at the file level (filesystem data) or as a database entity.
logical unit number (LUN)	Last part of a SCSI address (channel, target ID, LUN). LUNs are numbered 0 - 7.
low watermark (LWM)	Level to which HSM lowers filesystem utilization as the result of a demand staging run.
manual backup	Schedules backups from the command line using the ebbbackup command and the names of one or more work items or work groups.
maximum concurrent backups	Backup configuration parameters for limiting concurrent processing of work items at various points in the system. There is such a parameter for: the server software as a whole, each trail, local client software, and each remote client software (which applies to network backup of filesystems only).
media duplication	Feature of the EDM server that enables you to create a duplicate set of backup media automatically after each backup session.
media list	Displays information for each volume that the Library Unit Manager contains. By default, the media list includes the library unit name, drive, slot number, and volume information (name, barcode, and sequence number), and status of volume scheduling.

media rotation	For each trail, a new tape cartridge is started at the beginning of a new rotation period. This way, a set of tapes is created for each rotation period that just holds data from that period of time. Each such instance of the trail is called a media rotation.
media set	Set of media. See trailset.
media type	Type of physical storage medium such as digital linear tape (DLT) cartridge.
meta key	User-definable key that you can map to any key on your keyboard. Most X window applications include a default mapped Meta key. Use the xmodmap command to display the key map for your keyboard.
migrated data	Data that HSM has moved from one location to another. Other terms used are staged data or staged image.
migration	Process of automatically or manually moving data from magnetic disk to secondary storage. Migration is synonymous with staging.
migration server	EDM with HSM option. Contains the client, work item, media, schedule, and server configuration information. In addition, the Migration Server contains specific watermark information on when to stage files out from local storage to other storage media (e.g., optical or tape storage).
network backup	Backup of filesystems over the network, using core EDM Backup functionality. Backup of data designated by filesystem work items.
network migration client	Workstation or fileserver on the network that has data managed by EDM Migration software.

obsolete work items	After autoconfiguration, work items (filesystem work items) that no longer validly designate a current filesystem or a raw partition. Backup of these work items will fail if they are not moved out of a work group and assigned to a template.
offline_0	Offline Library Manager (offline_0) keeps track of volumes that are located outside a physical library unit, usually in a nearby storage rack or shelf. A volume logically enters the offline Library Manager when you eject a volume from a physical library unit.
offsite backup	Concept of storing backed-up data in a location outside of the building boundaries or the EDM Backup server.
offsite_0	Volumes that are located in the offsite Library Manager (offsite_0) represent those volumes that are located beyond the building's boundaries, such as an offsite archival location. A volume logically enters the offsite Library Manager when you eject a volume from a library unit or the offline Library Manager.
optical disk	Less expensive storage medium that uses laser technology to read and write data. Two types of optical disks are: <i>WORM</i> (write once read many) disks and <i>EO</i> (erasable optical) disks.
PC work item	Specifies the PC (NetWare, Windows NT, or OS2) or OpenVMS client data you want to back up. You can specify which directories and files to include or exclude for backup. See also work item.
periodic staging	Scheduled filesystem staging runs that are set via crontab to return disk utilization to the low watermark. Periodic staging is another name for bulk staging.

- pop-up menu** Menu that opens when you place the pointer over an object and click mouse button 3. The pointer changes to an icon if a pop-up menu is available for that object. Pop-up menus appear in the browser of the Main window and the Library Units and Drives area of the Library Unit Manager window.
- port control** Allows you to control the TCP ports used by the EDM to communicate with clients on the other side of a firewall.
- prestige reserve** Used for files that have been staged out, but also remain on the system's magnetic space. This magnetic space can be released quickly if disk utilization crosses the HWM. To allow filesystem usage to return to the LWM during a demand-staging event, the prestige reserve is typically the same size, or slightly larger, than the green zone. See also *working set*.
- prestige watermark (PSWM)** Predefined level at which HSM begins to stage files out to secondary storage. The prestaged files still reside on magnetic disk. See also low watermark (LWM) and high watermark (HWM).
- prestaging** Process in which files are written to secondary storage but their space on magnetic disk is not released. To minimize staging delays, HSM anticipates staging requirements and prestages additional files to allow the magnetic filesystem utilization to be lowered quickly if the HWM is crossed or if a demand staging run is necessary.
- primary storage** Main location, usually magnetic disk, for filesystem data storage. Magnetic disk storage is an example of primary storage and digital linear tape (DLT) is an example of secondary storage.

primary trailset	Because of the alternate night scheduling feature, it is possible for each template to have two trailsets. Therefore, even when you do not use alternate night scheduling and there is the only trailset, you will see it labeled Primary trailset . In addition, the name of the default trailset is "primary".
random setting	Sample watermark setting intended for filesystems where reads outnumber writes, but where the access pattern is random and least-recently-used caching is ineffective. This would be the case, for example, in a government records office, where several files must be read in from staging media in order to analyze a new file. When the analysis is completed, there is no need to keep the files on magnetic disk, because the files won't be accessed again for an undetermined period of time. You can select this setting for filesystems that match this random data access pattern. Other sample watermark settings include, archive setting and cached setting.
raw partition	Special file, located in /dev, that acts as pointer to a device driver. It associates a location, type, and access mode with a physical device.
red zone	Area on magnetic disk reserved for processes with root privilege. This area is used to expand system log files and other system files when the filesystem is full. The red zone only exists on systems that support minfree.
Reliability Agent Scanner Daemon (RASD)	Functionality that actually monitors the EDM system; it includes the rasd script and rasd configuration files.
remote client software	Software located on client which performs network backup of filesystems on the client. Windows NT, NetWare, and OS/2 backup client software must be installed on the client platform. See also local client software.

- Remote System Monitor (RSM)** Software that polls the rasd_alert file, notifying Customer Service Database when any significant events are detected.
- Report window** Part of the EDM graphical user interface that allows users to run reports on the local EDM or on many EDMs in a designated domain. Reports can be set to run automatically and can be sent to a printer, a file, or an e-mail address. Started from the EDM Main window.
- restore** Copies a client's backup files from the backup server's media to the client's disk. See Restore window.
- Restore window** Graphical user interface for restoring backed up data. Started by selecting Restore from the EDM window or typing **edmrestore** on the server's command line. Also for administrator use, it is not limited to user-initiated restore from clients of their own files back to their own client. If you have administrator permissions, you can also change destination directories and clients. See also cross-client restore and root restore.
- root restore** Restore of any backup files or database information belonging to any user on the client to any directory on the same client. Suitable for a system administrator of a single host. See also cross-client restore and self-service restore
- rotation period** Backup configuration parameter in the schedule template. The period of days during which a full backup will be performed for each work item covered by a template (for automatic scheduling). For custom scheduling, it's the schedule period.

saveset record	Data that is saved on backup media from a single backup of a single work item. A saveset record contains the template name, work item name, the backup level, start and completion times, expiration times, and the backup trail. The saveset record is used to find the volume containing the backup data and the associated backup <i>catalog</i> .
schedule	In backup configuration, a template is set up to specify the scheduling of backups and the use of media. See <i>template</i> .
schedule period	Backup configuration parameter in the template. It's the rotation period for automatic scheduling. For custom scheduling, it's the number of days in the custom schedule. See also <i>rotation period</i> and <i>custom scheduling</i> .
secondary storage	Storage medium, such as a DLT cartridge, used as a backing store for the filesystem. See also <i>primary storage</i> .
self-service restore	User-initiated restore from UNIX clients of their own files back to their own client and directory. See also <i>cross-client restore</i> and <i>root restore</i> .
server	See <i>backup server</i> or <i>migration server</i> .
server software	EDM Backup, Volume Management, and optional HSM software located on the EDM server. This software configures, initiates, and controls backups, restores, and data migration. See also <i>remote client software</i> and <i>local client software</i> .
stage in	Movement of data from secondary storage to magnetic disk.
stage out	Movement of data from magnetic disk to secondary storage.

stageable filesystem	Filesystem configured for migration. A stageable filesystem is sometimes referred to as a “filesystem under migration control.”
stage-in daemon	Process that stages in files or deletes bitfiles when a staged file is modified.
staging	In HSM, the process of moving files from one level in the storage hierarchy to another; for example, from local storage to the EDM server or from magnetic disk to optical disk. Also referred to as <i>migration</i> .
staging targets	Destination for migrated data. Destinations include volumes, staging media, staging devices, and stores.
staging template	File that contains staging parameters for one or more filesystems. Each filesystem is associated with one staging template and each staging template can be used by several filesystems.
staging trail	One or more volumes that contain staged data for a particular filesystem or group of filesystems. Initially, a staging trail consists of one volume and grows to several volumes to accommodate the staged data. A staging trail and <i>staging template</i> share the same name.
stale data	Data that remains on optical disk or tape after the data is staged in to magnetic disk and modified. When a large number of files become stale, the volume becomes a good candidate for compaction.
store ID	Unique code that identifies a client store on the network.

System Monitoring Support (SysMon)	Software which conveys the functionality that provides system monitoring, including RASD (Reliability Agent Scanner Daemon) and RSM (Remote System Monitor).
target ID	Middle part of a SCSI address (channel, target ID, LUN). Target IDs are numbered 0 - 7.
template	<p>Also called backup schedule template. A set of specifications for scheduling backups and use of media. Each template is uniquely named and includes a list of work group names, the name of the trailset (which defines media use), and scheduling parameters, including the rotation period for scheduling full backups, weekend backup policy, and weeknight backup shift lengths.</p> <p>See also volume template.</p>
thrashing	Unnecessary staging of files and movement of the autochanger in a library unit.
trail	Serial set of volumes of a particular media type. Each trail of volumes is written to over the course of one rotation period and then a new set of volumes is started. The trail specification also defines how long to save the backup data and its associated online catalogs and records. While uniqueness is provided by the combination of template, trailset, and trail names, it is helpful to give a unique name for each trailset instance.
trailset	Trail or set of trails to which the backup data is written, constituting a complete set of full and incremental backups for a rotation period. While uniqueness is provided by the combination of (schedule) template, trailset, and trail names, it is helpful to give a unique name for each trailset instance. Also called media set.

username	Username on the client that is used by EDM Backup to execute the client software processing. (The username is “ebadmin” by default for all UNIX filesystems.)
volume	Secondary storage media, such as tape cartridges, that contains a volume label and is entered into the volume catalog.
volume catalog	File that contains information about all removable media that is known to the EDM server. The volume catalog holds detailed information for each volume including a unique volume identifier, media characteristics, and the volume’s current state.
volume ID	Unique identification number electronically assigned to every piece of secondary storage media managed by the server.
volume label	Unique machine-readable code written on the backup media that includes a <i>volume sequence number</i> , <i>volume state</i> , <i>volume ID</i> , and <i>volume name</i> . All forms of removable media must be labeled before volume management can allocate them to an application.
volume life cycle	Stages through which media (magnetic tapes and optical disks) pass in the EDM system. New media begins as unlabeled and moves to available (ready for general use for any trail) or allocated (ready for a particular trail only) depending on which template you choose during the labeling process. Other states include: foreign (non-EDM media), uncataloged (labeled on another EDM server), erasing (optical disks only), and expired (no longer writable, just readable).
volume management	See EDM Volume Management.

Volume Manager	Central volume management process within EDM software that manages all removable media known to an EDM server. The Volume Manager maintains the volume catalog and manages volume allocation, access, and volume life cycles.
volume sequence number	Unique identification number electronically assigned to every form of removable media managed by EDM Volume Management.
volume state	Specific mode or phase in the media's life cycle. See also volume life cycle.
volume template	Named template that volume management uses for labeling volumes. See also volume label.
watermark	<p>In HSM, preconfigured levels that divide a filesystem into disk utilization zones. Watermarks are expressed as percentages of total disk space. EDM Migration has three watermarks: high watermark, low watermark, and prestage watermark. The watermarks define the yellow zone, green zone, and prestage reserve.</p> <p>There are three sample settings: archive setting, cached setting, and random setting.</p>
window manager	Software that enables manipulation of windows.
work group	Set of work items that are to be backed up to the same set of media. Each work group is uniquely named and includes a list of like work items (that is, you cannot mix filesystem, PC, and database work items in the same work group).

- work item** Client resource that you want to back up. A resource can be a UNIX filesystem, data on a PC server, or an Oracle, Sybase, or Informix database. Each work item is uniquely named and specifies the filesystems, database, or PC data to be backed up. You cannot mix filesystem, database, and PC work items in one work group.
- working set** In HSM, the space on magnetic disk between the LWM and the non-stageable data. The working set represents the files that are accessed in a given period of time.
- WORM optical disk** Optical disk that does not allow data to be erased or rewritten.
- yellow zone** Space on magnetic disk between 100% capacity and the HWM. The yellow zone is reserved for processes to use while HSM brings filesystem usage back down to the LWM.

Index

A

- Access Control List (ACL) 5-8
- activity checklist 13-33
- adding library units or devices 17-1
- allocation. *See* volumes
- alternate network backups
 - online database 6-9
- alternate trailset 3-17, 19-5, B-13, B-59, B-70, B-75, B-85
- archival filesystems 11-15
- archive life, of tape and optical media 11-20
- archiving EDM system logs 2-7
- ATL StorLink support 1-3
- authorized backup list B-5, B-18, B-19, B-37, B-52
- authorized recovery lists B-5, B-19, B-20
- autochanger C-17
 - SCSI address C-18
- AUTOCONFIG 17-8
- autoconfiguration
 - client 3-7
- automatic scheduling 5-5, B-82
 - alternate trailsets B-13, B-76
 - backup shift B-14, B-84
 - computing B-74
 - crontab 14-2

- forcing baselines to run first. *See* baseline
 - backups, forcing before levels 0-9
- full during weekend rotations B-83
- initial client backup B-86
- level maps B-46
- load balancing B-43
- mutual exclusion with custom scheduling
 - B-86
- options B-82
- rotation periods B-74
- setting up the initial backup B-15
- specifying a baseline with B-11
- standard rotations B-14, B-83
- starting rotations B-73
- turning on B-14, B-83
- weekend rotations B-14

- automounted files
 - timeouts D-9
- autoscheduling 3-7, B-14

B

- backup
 - automatic 14-4, B-4
 - baseline reports 16-4
 - checking completion of 2-3

- backup (continued)
 - client home directory A-13
 - coordinating with stageout schedules 13-4
 - crontab 14-4, B-4
 - executing 14-4, B-4
 - filesystem 5-7
 - high priority 8-11
 - incremental 5-7
 - installation files A-2
 - nightly 14-4, B-4
 - processing 14-2
 - queries 16-3
 - reports 3-21
 - See also* reports
 - reuse of baseline volumes 12-18
 - schedule 11-22, B-82
 - starting 14-4, B-4
- backup account B-5, B-17
- Backup Activity Wizard 1-15, 3-5, 5-5, 14-2
- backup administrator
 - recovering files B-19
 - recovery administrator B-23
 - user names B-5, B-17, B-79
- backup catalogs 14-9
 - changing a filesystem B-30, B-35, B-49
 - changing a work item B-30, B-35, B-49
 - compressing B-11, B-58, B-69
 - creating 3-11
 - deleting incomplete 2-8
 - delta level B-11
 - distributing among several disks 10-10
 - expiration period 3-12
 - expiring 3-12, 10-4, B-66, B-67, B-68
 - locating during recovery B-68
 - processing of 20-31
 - recreating 10-2
 - retention period B-11, B-68
 - setting processing schedule 14-9
- backup client cleanup command B-9
- backup client directories A-13
- backup client initialization command B-9
- backup clients B-5
 - backing up 5-5
- backup completion reports B-13, B-71, B-79
 - locate failure information B-80
- backup completion script B-79
- backup configuration file B-1 to B-87
 - backup template fields B-70
 - editing rules B-3
 - field summary B-5
 - findxcpio macros D-3
 - server B-5
 - server block fields B-16
 - server-level configuration fields B-5
 - specifying backup files D-7
 - trailset fields B-58
 - work item fields B-32
- Backup Configuration window 1-17, 3-13, B-1
- Backup Configuration Wizard 3-13, 14-2
- backup coverage reports 16-32
- backup data B-70
 - definition of 3-20
 - expiring B-65, B-66, B-67
 - locating during recovery B-68
 - retention period B-65
 - storage of 3-11, 5-12
- backup databases
 - when to back up 2-10, 13-7
- backup duplicate reports 16-12
- backup expiration 5-19, 5-20
- backup failure reports B-13, B-80
- backup failure script B-80
- backup levels 11-34, B-7, B-8, B-10, B-34, B-38, B-45
 - alternating trailsets B-76
 - autoscheduling 3-7
 - baseline level written to a trailset B-11, B-64
 - consolidating catalogs B-11, B-69
 - custom schedule in eb.cfg B-15, B-82
 - custom scheduling B-82

- forcing baselines to run first B-13, B-43, B-70, B-80 to B-81
- forcing level 0 B-30, B-35, B-49
- full 3-7, 3-9, 3-11
- incremental 3-7, 3-11, 5-5
- load balancing B-7, B-43, B-54
- mapping B-8, B-45, B-56
- retaining backup catalogs B-68
- retaining saveset records B-11, B-69
- specification in eb.cfg B-84
- specification of the filespec B-37, B-52
- specifying B-61
- specifying the filespec for levels B1 and B2 B-7, B-38
- time between full backups B-12, B-74
- trails B-11
- work item options by B-34
- writing a trail B-58, B-60
- backup path
 - versus restore path 5-16
- backup priority B-7
- Backup Report window 16-1
- backup reports 16-1
 - See also* reports
- backup saveset 3-20
 - backup catalog 3-20
 - backup data 3-20
 - saveset records 3-20
- backup schedule B-35, B-42, B-72
 - specification in eb.cfg B-82
- backup server
 - backing up all filesystems B-38
 - backing up database files B-42
 - database files B-46
 - directories A-2
 - maximum simultaneous client backups B-5, B-60, B-62
 - name B-5
 - priorities and resources B-42
 - software files A-2
 - backup server log file. *See* template log files
 - backup shifts B-31
 - definition B-84
 - nightly 3-10
 - weekday goal B-14, B-82, B-84
 - weekend goal B-14, B-82, B-84
 - backup templates B-59
 - backing up work groups B-31
 - backup completion reports B-79
 - efficiency with fewer B-60
 - forcing baselines first B-43
 - list of fields B-12
 - mapping levels B-8, B-45, B-56
 - naming B-12, B-71
 - obsolete work items B-35
 - removing obsolete B-72
 - specification in eb.cfg B-70, B-84
 - backup trailsets 3-3
 - backup work groups 3-3
 - backup/HSM tag 11-9, B-34, B-39
 - backupdates file 11-9
 - backups 14-5
 - automatic. *See* nightly backup processing
 - Backup Activity Wizard 3-5, 5-5, 14-2
 - baseline work items 16-27
 - command-line 14-7
 - configuration window 3-13, B-1
 - configuration wizard 3-13, 14-2
 - coverage reports 16-32
 - database 2-10
 - deleting expired catalogs and media 2-8
 - deleting incomplete catalogs 2-8
 - disabling client backups B-19
 - executing from crontab. *See* nightly backup processing
 - expiring 5-19, 5-20, 10-3
 - expiring catalogs B-68
 - expiring data B-66, B-67
 - expiring saveset records B-68
 - from crontab 14-5, B-42

backups (continued)

- history
 - work items 16-16
 - interleaving on media 5-12
 - load balancing 3-7
 - local client 5-2
 - log files. *See* log files
 - manual 3-22, 14-7
 - modified files 3-11
 - network 16-37
 - NFS timeouts D-10
 - nightly processing 3-10
 - online 5-7
 - priority 5-4, 5-5
 - rotation period 3-7
 - scheduling 3-3, B-41
 - automatic. *See* autoscheduling
 - custom. *See* custom scheduling
 - from command line B-1
 - storing 3-11, 5-12
 - verifying backups 2-8
- backups.log files 5-15, 16-5, 16-35, B-27, B-28
- client B-6
 - server B-6
- barcode
- configuration C-15
 - inventory 8-21
- baseline
- backups
 - updating the saveset-to-baseline relations database with A-10
 - media 11-29
 - reports. *See* reports
- baseline backups
- active volumes 12-19
 - as backup levels. *See* backup levels
 - completeness option with B-8, B-34, B-45
 - custom scheduling B-85
 - definition 11-33
 - expiring saveset records for B-69

- filespec for B-7, B-34, B-38
 - forcing before levels 0-9 B-13, B-43, B-70, B-80 to B-81
 - level mapping, and B-47
 - level written for a trailset B-11, B-64
 - recreating automatically B-14, B-71, B-81
 - trails used for B-60
 - updating the saveset-to-baseline relations
 - database with 5-3
 - with stage-to-tape 11-21
- baseline filespec B-34
- how to specify B-38
- baseline volumes
- deallocating 5-20
- baseline-relative backups 5-20
- bitfiles 12-13, A-24
- 16-digit hexadecimal name 12-13, A-24
 - bitfile ID 12-13, A-25
 - description of 11-7
 - names 12-13, A-25
 - stale 11-30, 13-30
- buckets 12-7
- bulk staging 11-2
- automating 13-2
 - stage in 13-9
- byte array. *See* bitfiles

C

- cached filesystems 11-15
- candidate list 11-19, 12-5
- catalog disk subsystem 1-9
- catalogs 3-11
- backup. *See* backup catalogs
 - distributing among several disks 10-10
 - expiring 10-3
 - expiring backups 3-12, 10-3
 - processing 20-30
 - recreating backup 10-2
 - status of processing 16-9

- volume 7-4, A-20
 - volume template A-20
- checklist for port control 4-10
- circular log files 15-4, A-19
- cleaning cartridges
 - default barcode 8-10
 - injecting into library units 8-10
 - maximum usage count 8-10
 - usage count 8-11
- cleaning tape drives 2-9
- client account B-5, B-17
- client agent. *See* **emsd**
- client backup username B-5, B-17
- client directories A-13
- client log files 16-37
- client pacing 5-11
- client recoveries
 - authorizing B-5, B-19, B-20, B-23
- client rotations B-12, B-74
- client software 3-2, 5-6
- client stores 11-4, 11-6
 - bitfiles 12-13, A-24
 - changing name of 13-16
 - changing ownership of 13-16
 - clearing incomplete bitfiles 2-9, 13-31
 - creating 13-32
 - directory tree 12-12, A-23
 - future needs 11-9
 - how many to create 11-7
 - ownership 11-7
 - removing 13-32
 - when to back up 13-7
- client/server architecture 3-2
- clients
 - ~ebadmin login A-13
 - autoconfiguring 3-7
 - backing up 3-11, 5-5
 - backing up filesystems 3-6
 - backup installation A-13
 - client pacing 5-10, 5-11
 - database 5-4
 - disabling backups B-19
 - IBM ACLs 5-9
 - local backup 5-2
 - maximum simultaneous backups B-5, B-24, B-62
 - maximum simultaneous backups per trail B-11, B-60
 - multiple networked B-40
 - NetWare 5-4
 - new
 - scheduling first backup B-15
 - NFS filesystems D-10
 - platforms A-6
 - restoring data 3-12
 - unavailable during backup 5-5
 - work groups. *See* work groups
 - work items. *See* work items
- command-line interface 14-7, 18-1
 - See also* scripts, manpages
- command-line scheduling B-65
- commands
 - backup
 - eb_server_config** B-17
 - ebbackup** 2-8, B-4, B-17, B-41, B-71, B-81
 - ebimport** 10-14
 - ebrestore** 5-16, B-24, B-28
 - findxcpio** 5-7, B-7, B-8, B-10
 - startfind** 5-4, 5-7
 - recover
 - ebrestore** B-20, B-22
 - findxcpio** B-37
 - restore
 - edmrestore** 5-18
- compaction 13-6
 - administering 11-28, 13-29
 - baseline 11-29, 13-30
 - baseline volumes 12-18
 - coordinating with backup and HSM 13-5
 - dbreport** 13-19

- compaction (continued)
 - description of 2-6, 11-27
 - disabling 13-26
 - emxattr** file 11-27
 - goals 12-14
 - limits 13-7
 - manual 11-27, 13-28
 - report 11-30
- completeness option 5-5, B-7, B-34, B-45
 - defining 11-34, B-44
 - staged files, with B-43
 - with baseline-relative backups B-45
- completion reports 16-4, 16-29
 - See also* backup completion reports
- concise log 15-3
- conf_ism_staging_template**. *See* **emstconf**
- configuration
 - autochanger C-17
 - barcode C-15
 - client 3-7
 - database (HSM) 12-2, A-21
 - default backup 3-7
 - default trail 3-7
 - EDM Symmetrix Connect 6-15
 - logical unit number C-17
 - of Library Managers 17-1 to 17-14
 - watermarks 11-10
- configuration files
 - backup B-1 to B-87
 - See also* backup configuration file
 - eb.cfg A-8
 - Library Manager C-9
 - lm.cfg A-19
 - server block fields B-16
 - syslog 15-6
 - Volume Manager C-2
 - work group fields B-31
- connection
 - connection via B-7, B-40
 - use connection method B-9, B-57
- convenient property 13-10, 13-24
 - stageout 11-23
- coordinating HSM and backup 11-22, 13-5
- coverage reports 16-3, 16-32
- crash files
 - deleting 10-4
- creating
 - space on magnetic disks 10-13
- cron** 2-6, 13-31
 - autocompaction 12-15
 - backup process 3-10
 - message logging system 2-4
 - periodic staging 11-22
 - See also* command-line backups, crontab, nightly backup processing
- crontab 14-5, 16-26
 - adding entries from the EDM GUI 2-6, 14-4
 - backup 14-4, B-1, B-4, B-38, B-42
 - backup catalogs 14-10
 - file 14-4, 20-5, 21-2
 - message logging system 15-2
 - nightly backup 14-2
 - reports 16-1
- crontab file
 - deleting existing entries 2-9
- cross recovery B-5, B-20
- current staging volume 11-4, 13-19
- custom schedule 3-17
- custom scheduling B-65
 - alternate trailsets B-76, B-77
 - establishing the schedule B-15, B-84
 - executing backups 14-6
 - forcing baselines to run first. *See* baseline backups, forcing before levels 0-9
 - level maps B-46
 - mutual exclusion with automatic scheduling B-86
 - options B-82
 - rotation periods B-74, B-86
 - starting rotations B-73
 - turning on B-15, B-84

D

- daemon processes 12-9
- daemons
 - ebbackupd** 14-9
 - ebcatalogd** 5-3
 - Library Manager 8-4, A-19
 - notd** 8-4
 - Volume Manager 8-4, C-5
- daily log
 - distribution list 2-4
 - syslog 15-3
- daily tasks 2-2
- damaged disk, replacing 21-1
- database
 - backup levels B-47
 - EDM Symmetrix Connect clients 6-6, 6-7
 - EDM Symmetrix Path 6-2, 6-12
 - EDM Symmetrix Path clients 6-3, 6-6, 6-7
 - files 2-10
 - network 6-2
 - online network 6-2
 - online network clients 6-3, 6-6, 6-7
- database backups 6-2
 - backing up client 3-11
 - restoring 5-18
- dbreport** 11-27, 11-30, 13-19, 13-28, 16-34
 - compaction report 13-7
- deadlock potential
 - with stage-to-tape 11-21
- debug logging level 15-3, B-77
- debug mode
 - volume management C-6
- deconfiguring a Library Manager 17-14
- default cleaner barcode 8-10
- default configuration
 - backup 3-7
 - trails 3-7
- default_template.log file 16-36
- delay factor 11-22
- deleting expired catalogs, media, and saveset
 - records 2-8
- delta catalog 10-8
- delta inventory 8-21
- delta level. *See* backup catalogs, compressing
- demand stageout 11-2, 11-12
- detail log 15-3
- device drivers
 - adding 17-1
 - installing 17-4
 - removing 17-7
 - updating 17-6
- device node C-17, C-18
- df** 13-18
- directories
 - backing up client 3-11
 - client A-13
 - ~ebadmin A-13
 - bin A-16
 - EB A-15
 - EB_DB A-16
 - home A-13
 - man A-16
 - HSM A-21
 - server
 - bin A-5
 - catalogs A-5
 - client A-6
 - config A-8
 - db A-9
 - locks A-10
 - man A-10
 - preconfig A-10
 - VM A-2
- disabling staging 13-24
- disaster recovery 20-1 to 20-33
 - an example 20-2
 - client restore 21-1 to 21-6
 - disabling backup commands 20-5, 21-2
 - hardware 20-6

- disaster recovery (continued)
 - HSM local configuration 20-18
 - library manager configuration 20-11
 - preparation for 19-1 to 19-6
 - restore LOCAL_DATABASE 20-15 to 20-22
 - software 20-6
 - strategy 19-2
 - using duplicate volumes 20-21, 20-23
- disaster reports 2-5, 16-4, 16-19 to 16-26, 19-4
 - disk configuration 20-7
 - installation report 20-9 to 20-11
 - vfstab 20-8
- disconnecting library units 17-14
- disk capacity
 - monitoring 10-5
- disk crash
 - client recovery 21-2
- disk thrashing B-81
- disk thrashing, preventing. *See* exclusion tags
- disks
 - backing up magnetic 3-10
 - capacity 10-5
- DLT media 8-27, C-18
- do not load balance field B-7, B-34, B-43, B-54
- drive cleaning 8-22
- drive preemption 8-16
- drives
 - busy 8-11
 - contents file A-19
 - preempting 8-16
 - scheduling 8-16
 - SCSI address C-18
 - See also* tape drives
- DTF 8-27
- duplicate command options 16-13
- duplicate volume sequence numbers 8-26
- duplicate volume states 9-28
- duplicate volumes 7-5
 - in disaster recovery 20-23

- duplication state 16-12
- duplication status 16-12

E

- eb_server_config** 5-14, 20-11, 20-13, B-17
- ebadmin client backup user name B-5, B-17
- ebadmin login A-13
- ebbackup** 2-8, 5-3, 5-4, 5-13, 5-14, 10-3, 13-7, 14-7, 21-2, B-4, B-17, B-41, B-71, B-81
- ebbackupd** 5-3, 5-4, 5-12, 5-14, 5-15
- ebcatalogd** 5-3, 5-13, 14-9, 20-16, 20-22
- ebcatclean** 2-8, 21-2
- ebcat.log file 16-5, 16-36
- ebcatproc** 20-31
- eb.cfg file A-8
 - backup. *See* backup configuration file
- ebcp** 13-14 to 13-18, 13-27, 13-32
- ebcrecover** 5-17, 18-3
- ebexpire** 2-8, 10-4, 10-14, 12-19, 21-2
- ebfs_import** 13-15
- ebimport** 3-21, 10-14, 20-16
- ebrecover** 5-17, 18-3
- ebreport** 9-25, 10-8, 16-4 to 16-35
 - backup 16-6 to 16-9
 - baseline 16-27 to 16-29
 - coverage 16-32
 - disaster 2-5, 16-19 to 16-26, 19-4 2-8
 - duplicate 16-12
 - history 2-3, 2-8, 10-11, 16-15 to 16-18
 - media 2-8, 16-10 to 16-11
 - run from cron 2-8
- ebreport coverage** 16-32
- ebreport duplicate** 9-25, 9-27, 16-12, 16-13, 16-14
- ebreport history** command options 16-16
- ebreport media** 9-25, 9-29, 16-10
- ebrestore** 5-13, 5-16, 13-34, 13-35, 13-36, 14-9, 16-5, 18-2, 18-3, 18-5, 20-19, 20-28, 20-29, 20-30, 20-31, B-20, B-22, B-24, B-28
- ebserver field B-5, B-17

- EDM 1-14
 - graphical user interface (GUI) 1-12
 - hardware 1-9
 - internal components 1-9
 - overview 1-3
 - software 1-11
 - EDM Library Unit Manager 7-3
 - EDM Migration
 - determining backup status for clients 5-3
 - EDM Migration clients
 - completeness option with B-34
 - work-item options for B-34
 - EDM Symmetrix Connect 1-6
 - EDM Symmetrix Path 1-5, 6-2, 6-12
 - edm_services
 - files 4-14
 - edmcrestore** 3-12, 5-17
 - edmproc** 8-5, 8-7
 - edmremote** 1-13
 - em_make_cl** 12-4 to 12-6
 - embsi** 11-2, 12-7, 13-9
 - emcheck** 11-31, 13-12, 13-33
 - emchmod** 11-23, 11-25, 11-26, 12-7, 13-10, 13-24
 - emcompact** 2-7, 11-27, 12-14, 13-5, 13-28, 16-35
 - emdu** 13-18
 - emfmd** 12-7
 - emfsconf** 11-22, 12-2, A-21
 - emfsreport** 13-19, 13-20 to 13-22, 13-26
 - emls** 11-23, 11-24, 12-7, 13-2, 13-10, 13-18
 - emlsconf** 20-17
 - emmasterd** 12-4 to 12-5
 - emsccheck** 2-9, 13-31
 - emschs** 13-16, 13-31
 - emsd** 12-9
 - emsid** 12-7
 - emsmks** 13-32
 - emsmvs** 13-16
 - emsstat** 11-31, 13-32
 - emstage** 11-2, 12-7, 13-2, 13-9
 - emstconf** 12-2, A-21
 - emsundel** 2-9
 - emsysconf** 12-2, A-21
 - emvck** 2-7, 11-28, 13-5, 13-6, 13-29
 - emxattr** file 11-27
 - epcleanup** 2-8, 10-4, 10-15
 - epnewlog** 2-7
 - Epoch Bitfile System (EBFS) 8-3
 - Epoch Volume Management. *See* volume management
 - eptrunclog** 2-7
 - erasable optical (EO) media 8-27, C-18
 - erasing, volume state 7-7
 - error logs, rotating 13-8
 - error messages 15-1
 - format 15-5
 - logging level B-77
 - evmchvol** 8-11
 - evmclean** 2-9, 8-22
 - evmimport** 8-9
 - evminject** 8-10
 - evminventory** 20-19
 - evmlistd** 8-3, 8-26
 - evmstartup** 20-22
 - evmstat** 8-3
 - exclusion tags 5-5, B-7, B-34, B-40, B-53
 - expiration 3-20
 - backup 5-19, 5-20
 - backup catalogs 3-12, 10-4, 10-14
 - expired media 7-6
 - expiring backups 10-3
- ## F
- failed backups, reprocessing 14-7
 - failed duplications 9-19
 - removing from the queue 9-18
 - vmdup -remove 9-18
 - failure reports 16-2, 16-4, 16-31
 - See also* backup failure reports
 - features
 - EDM Symmetrix Connect 1-7

- fencepost 11-24, 12-6
- file control properties 11-23, 13-10
 - listing and changing 11-24
- file locking 11-23, 11-24, 13-12
- file names
 - /usr/epoch/man A-13
 - /usr/man A-13
 - ~ebadmin A-13
 - ~ebadmin/*client-name*/bin A-13
 - ~ebadmin/*client-name*/HISTORY A-13
- file properties
 - staging control 11-23
- file recovery 3-12, 5-16, B-16
 - catalogs B-68
 - commands. *See* commands, recover
 - completeness options, and 11-35, B-34, B-45
 - enabling B-5, B-19, B-20, B-23
 - logging B-28, B-29
 - recovering another client's files. *See* cross recovery
 - time vs. rotations B-74
 - using a baseline for B-14, B-81
- file recovery commands
 - ebrestore** B-24, B-28
- file serial numbers. *See* inodes
- files
 - automounted D-9
 - backing up client 3-11
 - backing up modified 3-11
 - circular log A-19
 - daily changes 10-7
 - deleting tmp and crash 10-4
 - determining number of backed up 10-9
 - drive content file A-19
 - eb_ci_data A-7
 - eb_ci_particulars A-7
 - eb.cfg A-8, B-1 to B-87
 - excluding from backup D-8
 - Library Manager configuration C-9
 - lm.cfg A-19
 - log 15-2, 15-4
 - recovering 5-16
 - recovering client 3-12
 - scanning client 5-7
 - specifying for backup D-7
 - specifying with findxcpio D-7
 - staging control properties 11-23
 - valid.dat A-19, C-15
 - volume management A-18
 - Volume Manager configuration C-2
- filespec B-8, B-34
 - specifying B-37, B-52
 - to back up B-37
- filesystem delay 11-22
- filesystems
 - archival use 11-15, 13-20
 - backing up client 3-6, 3-11
 - cached 11-15
 - cleaning up 10-4
 - delaying stageout 11-22
 - exceeding HWM 11-12
 - exceeding PSWM 11-11
 - general purpose use 11-15
 - limits 11-18
 - populating 13-23
 - random use 11-16
 - scanning 5-7
 - with many small files 11-19
- findxcpio** 5-3, 5-7, B-7, B-8, B-10, B-37
- findxcpio** macros D-3
- firewall 4-5
- Flags column 11-24
- foreign volumes 7-6
- freezing a client store 13-31
- full backups 3-7, 3-9, 3-11, B-63
 - avoiding extra B-7, B-43, B-54
 - performing initial B-15, B-86
 - rotations B-12, B-74

uncompressed catalogs B-70
 week or weekend B-14
 full filesystems 11-10, 13-19
fuser 15-4

G

graphical user interface (GUI) 1-14

H

hardcopy documentation xxxi
 hardware
 configurations 6-15
 replace after a disaster 20-6
 SCSI address 17-4
 hardware address configuration C-18
 high watermark 11-2, 11-18, 13-25
 history reports 16-4, 16-15 to 16-18
 HITC 8-27
 HSM
 activity checklist 2-2
 client installation 1-21
 Configuration window 1-22
 disabling 13-25
 performance factors 11-5
 procedures run via **cron** 2-6
 staging to tape 13-3
 support 1-8
 volume labels 7-8
 HSM backup tags A-10
 HSM directories A-21

I

I/O daemon (iod)
 circular log file 15-4
 importing volumes 8-9
 importing duplicate volumes 8-9
 media information 8-10
 incremental 5-7

incremental backups 3-7, 3-9, 3-11, 5-5, 5-7
 expiring catalogs B-68
 mapping B-48
 maximum concurrent clients B-63
 week day B-83
 inheritable properties 11-23, 11-25
 initiating manual duplication 9-8
 inject 8-11
 injecting cleaning cartridges 8-10
 inlet configuration C-15
 inlet types
 automatic 8-8
 manual 8-8
 inodes 11-19
 information about 13-18
 Install Wizard 1-15
 installation report 20-9
 interfaces 1-13
 recovery 20-22
 interleaving backups 5-12
 inventories
 delta 8-21
 library unit 8-20
 inventory methods
 verify barcode 8-21
 verify label and barcode 8-21
 inventory table 8-19
ism_compact command 21-2

K

keep backup catalogs B-68
 keep backup catalogs specification 10-3
 vs. rotation period 10-5
 keep backups specification 10-3, B-11, B-65
 keep property 11-23
 keep saveset records specification B-11, B-68
 kernel messages 15-5
 key processing concepts 3-3

L

labels

- volume 7-4, 8-12

level 0 backups. *See* full backupslevel 9 backup. *See* incremental backups

level map B-8, B-34

- precedence over schedule block fields B-83

- specifying B-45, B-56

levels. *See* backup levels

Library Managers

- circular log file 15-4

- configuration files C-9

- daemon 8-4, A-19

- deconfiguring 17-14

- drive preemption 8-16

- icon 7-12

- initial startup 7-13, 8-19

- listing configured 17-3

- names 17-3, C-15

- naming convention C-9

- offline 7-13, C-19

- offsite 7-13, C-19

- reconfiguring 17-1

- simultaneous backup example 8-17

- subdirectories A-18, C-10

- verifying priority 8-16

Library Unit Manager window 1-16

library units

- adding 17-1

- automatic inlet 8-8

- barcode support C-15

- contents of A-19

- deconfiguring 17-14

- drive address C-18

- drive content file A-19

- drive preemption 8-16

- drive scheduling 7-13, 8-16

- ejecting volumes 7-14

- inlet configuration C-15

- inventory 8-18, 8-20, A-19

- list 7-13

- operations 8-1

- SCSI address C-17

- slot contents C-15

life cycle management 7-4

limit throughput field B-6, B-25

LM_INLET_IGNORE_ON_OPEN C-19

LM_MAX_RESIDENT_TIME 8-17

LM_MIN_RESIDENT_TIME 8-17

lm.cfg file A-19

lmconfig 8-4, 17-1 to 17-17, 20-11, C-1

- AUTOCONFIG 17-8

- troubleshooting 17-17

load balancing 13-6

- excluding a work item from B-34

- excluding work items B-7, B-43, B-54

local account

- backup client user name B-5, B-17

local client B-32

- backing up 5-2, B-38

- connecting to the server 5-4

- level map for database files B-46

- log files 16-37

- priority for database files B-42

- using the migration backup tag on B-38

- work-item options B-34

LOCAL_DATABASE 20-2 to 20-3

- backup 19-3, 20-15 to 20-22

- late backup 2-10

- priority 2-10, B-42

- restore 20-15 to 20-22

locked property 11-23, 11-24, 13-12

- performance factors 11-24

locking files 13-11

log files 3-20, 16-5, 16-35 to 16-37

- backup 5-15

- backup template 16-36

- backups.log 16-5, 16-35

- circular 15-4, A-19

- client 16-37
- ebcat.log 16-5, 16-36
- expiring 10-4
- expiring oldest data B-27
- mntfault 15-3
- recoveries.log 16-5, 16-35
- server log file 16-5
- syslog 15-3
- template_name 16-5, 16-36
- volume management 15-1
- See also* reports
- log message format 15-5
- logging level 16-36, B-13, B-77
 - debug 16-36
 - per file 16-36
 - stats 16-36
- logical unit number C-17
 - autochanger C-17
 - in SCSI address 17-4
- low watermark 11-2, 11-10 to 11-18, 11-22, 13-19

M

- magnetic disks
 - backing up 3-10
 - capacity 10-5
 - creating space on 10-13
 - utilization of space 13-23
- magnetic tape drive. *See* drives, tape drives
- magnetic tapes
 - backup 3-7
- mailerr script 5-15, B-13, B-80
- mailing the daily log file 2-4
- mailok script B-13, B-79, B-80
- manpages 18-1, A-10, A-16
 - backup and restore 18-2
 - HSM 18-9
 - migration 18-9
 - volume management 18-6
- manual backups. *See* command line interface
- manual operations 3-22
- mapping backup levels. *See* backup levels, mapping
- master staging daemon 12-5
- maximum simultaneous clients B-5, B-24, B-60, B-62
- media
 - allocating volumes B-76
 - backup levels B-11, B-60, B-61
 - choices 11-19
 - conserving B-75
 - designated by a trail B-60
 - DLT 8-27
 - ejecting 8-14
 - erasable optical 8-27
 - expired 7-6
 - expiring B-66, B-67, B-68
 - importing volumes 10-13
 - inserting into library units 8-8
 - management B-61
 - maximum trails per media type B-6, B-24
 - maximum uses 18-6
 - reports 16-10 to 16-11
 - retention B-11, B-65, B-67
 - reusing B-65, B-67
 - rotations 16-10
 - tracking B-61
 - trailsets B-58, B-60
 - transferability 11-22
 - using trailsets B-11
 - WORM 7-11, 8-27
 - See also* volumes
- media duplication 3-18, 9-1, 9-31, 19-5
 - append mode 9-5
 - automatic 9-8
 - backup duplicate report 16-14
 - backup or restore 9-2
 - canceling 9-17
 - concurrent duplications 9-11, 9-24
 - Control window 9-20
 - determining duplicates 9-24

media duplication (continued)

- disabling 9-15
- duplication state 16-12
- duplication status 16-12
- Duplications window 9-30
- ebreport duplicate** 9-27
- ebreport media** 9-10
- evmstat** 9-11
- failed 9-20, 9-29
- importing a duplicate 9-29
- initiating 9-8
- manual 9-2, 9-8
- manually disabling 9-14
- media list 9-10
- mode 16-12
- new mode 9-5, 9-6
- offline volume 9-14
- padding block 9-4
- pausing 9-16
- process 9-2
- reenabling 9-14, 9-15
- reject mount request 9-29
- rescheduling 9-19, 9-25
- rescheduling a single volume 9-14
- rescheduling an offline volume 9-22
- resuming 9-17
- verifying status 9-10
- vmdup** 9-3, 9-8
- vmdupcfg** 9-3, 9-4, 9-12, 9-15
- vmdupd** 9-3, 9-9, 9-13, 9-16, 9-18

media rotation

- starting new B-74

media types C-18

media wear

- with stage-to-tape 11-21

message logging system 15-1

- default configuration file 15-6
- features 15-2
- rotating, archiving log files 13-8

message number 15-5

migration volumes

- verifying information on 13-6

migration, disabling 13-25

mntfault log 15-3

monitor

- active backups 3-22, 16-3

monitoring magnetic disk space 10-10

multiple networked clients B-40

multiplexed storage 3-13

N

names

- backup account B-5, B-17

naming conventions

- Library Manager C-9

NetWare

- work items
 - fields in configuration file B-9

network

- database vendors 6-3
- limiting the available bandwidth B-25
- multiple networked clients B-40
- server login name B-17

network backup of filesystems 16-37

network clients

- moving files between 13-16
- work item options B-34

network database 12-10

network database (HSM) A-22

network migration server 12-9

NFS timeouts D-10

nightly backup processing 3-10

NIS password map B-17

non-mirrored configuration 6-17

non-stageable filesystems 13-19

notify daemon (**notd**) 8-4

O

offline

Library Manager 7-14, C-19

offsite

Library Manager 7-14, C-19

media 19-5

storage

alternate trailsets, of B-70

online

backups 5-7

books 1-12

catalogs 3-11

database backups 6-7

help 1-12

online documentation xxx

online help xxx

optical disks

compacting 11-27

P

padding blocks 9-4, 16-12

pass count, of tape and optical media 11-20

performance factors 11-5, 11-19, 11-24, 11-31, 13-6

performing full backups B-15, B-86

periodic stage out 11-2, 11-22

automating 13-2

coordinating with backup 13-5

permissions 3-19

backup and recovery B-5, B-17

recovery

root permissions B-5

pid. *See* process ID

port control 4-1

checklist 4-10

clients 4-18

default port ranges 4-6

files 4-14

firewall 4-5

making changes 4-20

restrictions 4-4

the portservices CLI 4-3

portservices files 4-14

prestige 11-2, 13-9

description of 11-10

policy 11-15

reserve 11-13 to 11-15

watermark 11-10 to 11-15

preventing stage outs 13-11

primary trailset 3-17, 19-5, B-13, B-59, B-70,
B-75

priority B-34

backup 2-10, 5-5, B-7, B-43

controlling backup 5-4

settings B-42

work item B-41, B-54

process ID

in log messages 15-5

processes

Library Manager 8-4

Volume Manager (**vmdaemon**) 8-4

product description 1-7

PROM level 17-5

R

random filesystems 11-16

reattaching staged-out files 13-15

reconfiguration reboot 17-5

recover

administrator list B-5, B-23

client data 3-12

disabling B-20

recoveries.log file 5-15, 16-5, 16-36, B-6, B-29

recovering bitfiles 2-9

recovery

disaster 19-1, 20-1, 21-1

log files 5-15

recreating a baseline automatically B-14, B-71,
B-81

recreating backup catalogs 3-21, 10-2

- remote diagnostic modem 1-10
- remote GUI launch 1-13
- removing
 - library units 17-14
 - unnecessary files 10-15
- replacing a damaged disk 21-1
- Report window 1-18
- reports 16-1 to 16-34
 - backup 2-3, 16-4, 16-5
 - baseline 16-4, 16-27
 - completion 16-4, 16-29
 - coverage 16-3, 16-4, 16-32
 - failure 16-2, 16-31
 - history 16-4, 16-15 to 16-18
 - media 9-25, 16-10, 16-11
 - backup completion 5-14
 - backup failure 5-15
 - backups.log 5-15
 - disaster 16-4, 16-19 to 16-26, 19-4
 - filesystems not backed up 16-3
 - HSM 11-30
 - in crontab 16-1
 - installation 20-9
 - online 3-22, 16-3
 - recoveries.log 5-15
 - volumes 16-34
 - See also* **ebreport**, log files
- reserved files in VxFS 13-12
- residence priority 11-23, 11-25, 12-6, 13-11
- resident files
 - backing up B-8, B-32, B-34
- restage** 11-33, 13-13
- restore
 - cross-client 5-9
 - database backups 5-18
 - files with ACLs 5-9
 - how it works 5-16
 - server disaster 20-1
- restore modes 3-20
- restore path
 - versus backup path 5-16
- Restore window 1-19
- restoring client data 3-12
- Restricted by Application 8-25
- restricted volume template 7-8
- restricted volumes 11-4
- revision levels
 - Oracle 6-7
- robot 7-13
- robot. *See* autochanger
- rotating EDM system logs 2-7
- rotation period 19-5
 - 14 day B-12, B-75
 - 28 day B-75
 - 7 day B-74
 - alternating trailsets B-76, B-77, B-85
 - autoscheduled 3-7
 - custom scheduling B-15, B-86
 - default B-74
 - definition 3-15
 - initial backup B-15, B-87
 - load balancing B-43
 - media B-76
 - specifying B-12, B-70, B-74
 - vs.* keep catalogs period 10-5
- rotation schemes
 - full backups during weekend rotations B-82, B-83
 - standard B-82, B-83
- RPC
 - HSM protocol 12-8
- rvmoper 8-1

S

- saveset records
 - definition of 3-20
 - expiring 10-3, B-66, B-68
 - online 10-2
 - retention period B-11, B-68
- SBus cards 1-9

- scheduling
 - automatic 3-4, 5-7, 14-6
 - backup 3-3, 14-2, B-82
 - catalog processing 14-9
 - command line 3-5
 - custom 3-4, 3-17, 14-6
 - nightly start of backups 14-3
- scheduling backups B-14, B-41, B-42, B-46, B-71, B-72
 - alternating trailsets B-76
 - work groups B-31
- scheduling level 0 backup B-35
- scripts
 - epcleanup** 10-4
 - findxcpio** 5-7
 - lmconfig** 17-1 to 17-17
 - startfind** 5-4
- SCSI
 - address 17-4, C-17, C-18
 - bus slot C-17
- self-describing media 11-22
- server
 - concurrent client backups 3-6
 - database (HSM) A-22
 - server database files 2-10
 - software 3-2, 5-6
 - volume catalog A-20
- server database 12-10
- server directories A-2
- server log files 16-5
 - generated 16-5
 - See also* template log files
- shelf life, of tape and optical media 11-20
- software
 - restore after a disaster 20-6
- SPARCserver 1-9
- SRDF configuration 6-16
- stage-in daemon 12-7
- stage-to-tape 13-3
- staggering client staging runs 13-6
- staging
 - bulk or periodic 11-2
 - bulk stage in 13-10
 - candidate list generation 12-5
 - event-driven or demand 11-2, 11-12
 - file control properties 13-10
 - manual stageout 13-9
 - to tape 13-3
 - watermarks 11-2
- staging templates
 - description of 11-4
 - how many to create 11-5
- staging trails 11-4
- staging volumes
 - compacting 11-27, 13-28
 - verifying information on 13-6
- stale files 11-27, 11-29, 11-30, 12-7, 13-28, 13-29, 13-31
- standard rotations B-82
 - how to specify B-83
 - See also* weekend backups
- start trailset rotations B-12
- startfind** 5-4, 5-7
- startup parameter B-87
 - backup of new clients B-15
- stats
 - backup log level B-77
- store ID
 - associated with client store 11-7
- streams from database 6-8
- striped backups, tuning 6-8
- Symmetrix Connect backup
 - EDM 6-13
- Symmetrix non-mirrored configuration 6-17
- Symmetrix Path 1-5, 6-2, 6-12
- syslog messages 15-2
- syslog.conf file 15-1, 15-6
- system
 - activity log files 2-4

system console
 description 1-10
 messages 15-1
system logs
 newepochlog 2-7
 rotating, archiving, truncating 2-7
system monitoring
 dbreport 11-30, 13-19
 emfsreport 11-31 to 11-32, 13-19 to 13-23
system monitoring support GUI
 EDM Symmetrix Connect 1-20, 1-21
 online help 1-21
 SNMP 1-20
 Tivoli 1-20

T

tape cartridges. *See* magnetic tapes
tape drives
 mounting volumes into 8-12
 preempting 8-16
 scheduling 7-13, 8-16
tape library units 1-9
 controlling 7-13
 definitions 1-9
target ID, SCSI 17-4
template log files 16-36, B-13, B-14, B-78
template_name.log file 16-36
templates B-45
 backup 3-3
 baseline reports 16-27
 history reports 16-13, 16-16
 volume 7-8, A-20
 See also backup templates
thrashing 11-6, 11-21
tilde (~) ebadm A-13
timeouts
 NFS-mounted filesystems D-10
tmp files, deleting 10-4

trails 5-13, B-58, B-77
 allocating volumes to 8-23
 baseline 5-20
 default configuration 3-7
 definition of 3-8
 efficiency with fewer B-60
 maximum number per media type B-24
 specification in eb.cfg B-11
trailsets 3-3, B-31
 alternate 3-17, 19-5, B-13, B-59, B-70, B-75, B-85
 alternating between B-76
 definition of 3-3
 efficiency with fewer B-60
 fields B-9, B-11
 moving off site B-70
 name B-11, B-60
 primary 3-17, 19-5, B-13, B-59, B-70, B-75
 rotations B-12, B-73
 specifying B-58
 starting new rotations B-12, B-73

U

uncataloged volume 7-5
UNIX remote shell 5-4
unlabeled volume 7-6
unlabeled WORM 8-27
unrestricted volume template 7-8
unrestricted volumes 11-4
unverified volume 7-7
user ID
 in log messages 15-5
user names
 backup account B-5, B-17
user-level NFS daemon 12-4, 12-7

V

/var/adm/epoch_concise_messages 15-3
verifying priority in the queue 8-16
VERITAS 20-7

- virtual filesystem statistics 11-31
 - VM_ALLOW_DUP_SEQ_IMPORT 8-26
 - vmdaemon** 8-3, 8-5, C-5
 - failures 8-5
 - manual shutdown 8-5
 - See also* Volume Manager
 - vmdup** 9-3
 - vmdupcfg** 9-3, 9-15, 9-16
 - vmdupd** 9-3, 9-9, 9-15, 9-21, 9-23
 - valid.dat file A-19, C-15
 - valid.dat inventory file 8-19
 - volume allocation request 8-24
 - volume catalog 7-4
 - volume management
 - allocation, deallocation 8-1
 - circular log file 15-4
 - cleaning tape drives 8-22
 - debug mode C-6
 - directory structure A-2
 - eject media at the CLI 8-15
 - eject media through the GUI 8-15
 - lmconfig** 17-2
 - logs 15-1
 - overview 7-2
 - processes 8-2
 - re-injecting volumes automatically C-19
 - restarting 8-5
 - starting 8-1, 8-4
 - stopping 8-5
 - template catalog A-20
 - Volume Manager
 - configuration file C-2
 - daemon 8-4, C-5
 - directory 7-3
 - volume reports 16-33
 - volume sequence numbers
 - duplicate 7-5
 - volumes
 - allocation 8-23, 8-24
 - allocation request 8-24
 - catalog
 - location of A-20
 - compaction 2-6
 - containing backup data 16-10
 - current staging 11-4
 - dismounting 8-13
 - duplicate 8-26
 - duplicate sequence numbers 7-5, 8-26
 - erasing 7-7
 - foreign 7-6
 - inserting in a library unit 8-11
 - labeling 7-4, 7-8
 - labels, reading 8-12
 - life cycle management 7-4
 - maximum uses 18-6
 - media rotation 16-10
 - mount request 8-24
 - mounting 8-12
 - offline 7-14
 - reports 16-34
 - restricted by name 8-25
 - restricted or unrestricted 11-4
 - templates A-20
 - uncataloged 7-5
 - unlabeled 7-6
 - unrestricted 8-25
 - unverified 7-7
 - use 8-25
 - WORM 8-27
- VxFS
- reserved files 13-12
- W**
- watermarks (HSM) 11-10
 - window
 - Backup Configuration 1-17
 - Backup Report 1-18
 - HSM Client Installation 1-21
 - HSM Configuration 1-22

window (continued)

- Library Unit Manager 1-16
- Restore 1-19

work groups 3-3, 3-7, 3-16

- backing up B-12
- backing up to trailsets B-58, B-61
- backing up together B-73
- comprising work items B-31, B-32
- custom scheduling B-15, B-82, B-84
- fields B-6, B-8
- obsolete work items B-35
- specification in eb.cfg B-31

work items 3-3, 3-6, B-34, B-56, B-57

- backing up the local client B-38
- backup baseline reports 16-27
- backup failure reports B-80
- backup history reports 16-16
- changing B-35, B-51
- compressing catalogs B-70
- expiring backup catalogs B-68
- expiring backup data B-65, B-67
- expiring saveset records B-68
- fields B-6, B-8
- in work groups B-31, B-32, B-73
- list of fields by type of client B-34
- name B-7, B-8, B-10, B-36, B-51
- obsolete B-35
- order backed up B-7
- priority B-7
- specification in eb.cfg B-7, B-8, B-10, B-32
- specifying with findxcpio D-7

working set 11-13 to 11-15

- description of 13-20

WORM media 7-11, 8-27, C-18

WORM optical disks 8-27

Z

zones

- yellow, green, prestage 11-13